(19)日本国特許庁（ＪＰ）　(12) **公表特許公報**（Ａ）　(11)特許出願公表番号
特表2002－505451
（P2002－505451A）
(43)公表日　平成14年２月19日(2002.2.19)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード（参考）
Ｇ０９Ｃ　1/00	６５０	Ｇ０９Ｃ　1/00	６５０Ｚ	５Ｂ０６１
	６１０		６１０Ｂ	５Ｊ１０４
Ｇ０６Ｆ　13/36	３１０	Ｇ０６Ｆ　13/36	３１０Ａ	

審査請求　未請求　　予備審査請求　有　　（全 73 頁）

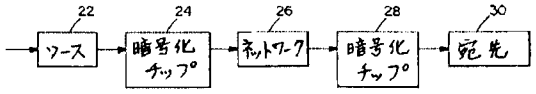
(21)出願番号　特願2000－533976(P2000－533976)
(86)（22)出願日　平成11年２月26日(1999.2.26)
(85)翻訳文提出日　平成12年８月25日(2000.8.25)
(86)国際出願番号　ＰＣＴ／ＣＡ９９／００１７６
(87)国際公開番号　ＷＯ９９／４４３２９
(87)国際公開日　平成11年９月２日(1999.9.2)
(31)優先権主張番号　０９／０３２，０２９
(32)優先日　平成10年２月27日(1998.2.27)
(33)優先権主張国　米国（ＵＳ）
(81)指定国　ＤＥ，ＧＢ，ＪＰ，ＫＲ

(71)出願人　モサイド・テクノロジーズ・インコーポレイテッド
カナダ、ケイ・２・ケイ　２・エックス・１　オンタリオ州、カナタ、ハインズ・ロード、11
(72)発明者　ジョーンズ，デイビッド・イー
カナダ、ケイ・２・ビィ　８・エス・５　オンタリオ州、オタワ、グレノン・アベニュ、424－1025
(74)代理人　弁理士　深見　久郎　（外５名）

最終頁に続く

(54)【発明の名称】　共用メモリ配線を有する暗号化プロセッサ

(57)【要約】
暗号化チップは、さまざまな秘密鍵および公開鍵の暗号化アルゴリズムを処理するようプログラム可能である。該チップは、演算処理装置のパイプラインを含み、該演算処理装置の各々は、秘密鍵アルゴリズム内の１ラウンドを処理することが可能である。データは、該演算処理装置間で、デュアルポートメモリを介して転送される。中央処理装置は、単一サイクルのオペレーションで、グローバルメモリからの非常に幅の広いデータ語を処理することができる。加算器回路は、比較的小さい複数の加算器回路を使用することによって簡素化され、合計およびキャリが複数サイクルでループバックされる。乗算器回路は、非常に幅の広い中央処理乗算器となるよう連結することができるように、より小さい演算処理装置乗算器を適用することによって、複数の演算処理装置と中央処理装置との間で共用することができる。



【特許請求の範囲】

【請求項１】　単一のチップ上に演算処理装置のアレイを含む、電子暗号化デバイスであって、各演算処理装置は、

暗号化アルゴリズムのラウンドを記憶するための命令メモリを含み、該ラウンドは命令のシーケンスを含み、各演算処理装置はさらに、

命令メモリからのラウンドを実現するためのプロセッサと、

暗号化データオペランドおよび該ラウンドの実行によって得られた暗号化されたデータを記憶するためのデータ記憶装置とを含み、

該アレイの演算処理装置は各々、ラウンドのうち１つを実現してその結果を連続する演算処理装置に転送し、それにより、該演算処理装置のアレイは演算処理装置パイプラインにおいて暗号化アルゴリズムの連続的なラウンドを実現する、電子暗号化デバイス。

【請求項２】　該データ記憶装置は、その１部分が、該線形アレイの隣接する演算処理装置間でデータを転送するために該線形アレイの隣接する演算処理装置間で共用される、請求項１に記載の電子暗号化デバイス。

【請求項３】　各演算処理装置は制御ユニットおよびＡＬＵを含み、該制御ユニット、命令メモリおよびデータ記憶装置はローカル演算処理装置データバスに接続され、該ローカルデータバスはスイッチによって２つの独立した区分に区分けされ、該制御ユニットおよび命令メモリは該区分のうちの一方に接続され、該ＡＬＵおよびデータ記憶装置は該区分のうち他方に接続される、請求項２に記載の電子暗号化デバイス。

【請求項４】　各演算処理装置は制御ユニットおよびＡＬＵを含み、該制御ユニット、命令メモリ、ローカルデータメモリおよび共用データ記憶装置はローカル演算処理装置バスに接続され、該ローカルバスはスイッチによって、該命令メモリおよび該制御ユニットを接続するローカル命令バス区分と、該ＡＬＵ、ローカルデータメモリおよび共用データ記憶装置を接続するローカルデータバス区分とに区分けされ、該スイッチは、該２つのローカルバス区分上で独立した同時動作を可能にするか、または、該２つのバス区分間の通信を可能にする、請求項２に記載の電子暗号化デバイス。

【請求項５】　各演算処理装置は、該演算処理装置内で乗算演算を行なうための乗算器をさらに含む、請求項４に記載の電子暗号化デバイス。

【請求項６】　該暗号化アルゴリズムの実現中に、該パイプライン内の各演算処理装置は、結果として得られたデータを、後続の演算処理装置が直接アクセスすることができるように該後続の演算処理装置と共用されるデータ記憶装置内に書込む、請求項２に記載の電子暗号化デバイス。

【請求項７】　該演算処理装置の共用データ記憶装置は、該線形アレイの隣接する演算処理装置間でデータを転送するために、該線形アレイの隣接する演算処理装置間で共用されるデュアルポートメモリで構成される、請求項２に記載の電子暗号化デバイス。

【請求項８】　各プロセッサは制御ユニットおよびＡＬＵを含み、該制御ユニット、ＡＬＵ、命令メモリ、ローカルデータメモリおよび共用データ記憶装置はローカル演算処理装置データバスに接続され、該ローカルデータバスはスイッチによって２つの独立した区分に区分けされ、該制御ユニットおよび命令メモリは該区分のうち一方に接続されかつ、該ＡＬＵ、ローカルデータメモリおよび共用データ記憶装置は該区分のうち他方に接続される、請求項７に記載の電子暗号化デバイス。

【請求項９】　各演算処理装置は、該演算処理装置内で乗算演算を行なうための乗算器をさらに含む、請求項１に記載の電子暗号化デバイス。

【請求項１０】　複数の演算処理装置の該乗算器は、より幅の広い乗算器の区分として連結されるよう適合される、請求項９に記載の電子暗号化デバイス。

【請求項１１】　各乗算器は部分積加算器を含み、該加算器は、独立した乗算器として動作しているときには第１の入力の組を選択し、かつ、連結されているときには、隣接する演算処理装置からの入力を含む第２の入力の組を選択するための入力選択回路を有する、請求項１０に記載の電子暗号化デバイス。

【請求項１２】　各プロセッサは制御ユニットおよびＡＬＵを含み、該制御ユニット、ＡＬＵ、命令メモリ、ローカルデータメモリおよび共用データ記憶装置はローカル演算処理装置データバスに接続され、該ローカルデータバスはスイッチによって２つの独立した区分に区分けされ、該制御ユニットおよび命令メモ

リは該区分のうち一方に接続され、かつ該ＡＬＵ、ローカルデータメモリおよび共用データ記憶装置は該区分のうち他方に接続される、請求項１に記載の電子暗号化デバイス。

【請求項１３】　グローバルランダムアクセスメモリおよびグローバルバスをさらに含み、データは該グローバルランダムアクセスメモリと該演算処理装置データ記憶装置との間で該グローバルバスを通じて転送される、請求項１に記載の電子暗号化デバイス。

【請求項１４】　該グローバルバスに結合された、該演算処理装置によって処理されるデータ語よりも幅の広いデータ語を処理するための中央処理装置をさらに含む、請求項１３に記載の電子暗号化デバイス。

【請求項１５】　各演算処理装置は、該演算処理装置内で乗算演算を行なうための乗算器をさらに含む、請求項１４に記載の電子暗号化デバイス。

【請求項１６】　複数の演算処理装置の該乗算器は、より幅の広い乗算器の区分として連結されるよう適合される、請求項１５に記載の電子暗号化デバイス。

【請求項１７】　各乗算器は部分積加算器を含み、該加算器は、独立した乗算器として動作しているときには第１の入力の組を選択し、また、連結されているときには隣接する演算処理装置からの入力を含む第２の入力の組を選択するための、入力選択回路を有する、請求項１６に記載の電子暗号化デバイス。

【請求項１８】　該中央処理装置は加算器を含み、該加算器は、

複数加算器区分を含み、該複数加算器区分の各々はキャリ出力および合計出力を有し、該複数加算器区分の各々は２つのオペランドのうち各オペランドの１区分を処理し、該加算器はさらに、

加算器サイクル内でキャリが得られる限り、連続的なクロックサイクル中、該キャリ出力を連続する加算器区分へのキャリ入力として選択するためのキャリ選択器と、

加算器サイクル内でキャリが得られる限り、連続的なクロックサイクル中、各合計出力を同じ加算器区分へのオペランド入力として選択するためのオペランド選択器とを含む、請求項１３に記載の電子暗号化デバイス。

【請求項１９】　各演算処理装置の各プロセッサは、ＭｍｏｄＮを計算するモジュロ調整演算を行なう、請求項１に記載の電子暗号化デバイス。

【請求項２０】　各演算処理装置の各プロセッサは、Ａ±ＢｍｏｄＮを計算するモジュロ加算または減算演算を行なう、請求項１に記載の電子暗号化デバイス。

【請求項２１】　各演算処理装置の各プロセッサは、Ａ×ＢｍｏｄＮを計算するモジュロ乗算演算を行なう、請求項１に記載の電子暗号化デバイス。

【請求項２２】　該暗号化デバイスは加算器をさらに含み、該加算器は、

複数加算器区分を含み、該複数加算器区分の各々は、キャリ出力および合計出力を含み、該複数加算器区分の各々は２つのオペランドのうち各オペランドの１区分を処理し、該加算器はさらに、

加算器サイクル内でキャリが得られる限り、連続的なクロックサイクル中、キャリ出力を連続する加算器区分へのキャリ入力として選択するキャリ選択器と、

加算器サイクル内でキャリが得られる限り、連続的なクロックサイクル中、各合計出力を同じ加算器区分へのオペランド入力として選択するオペランド選択器とを含む、請求項１に記載の電子暗号化デバイス。

【請求項２３】　単一チップ上に演算処理装置の線形アレイを含む、電子暗号化デバイスであって、各演算処理装置は、

暗号化アルゴリズムの少なくとも１つのラウンドを実現するのに必要とされるコードを記憶するための命令メモリと、

該命令メモリからの該ラウンドを処理するためのプロセッサと、

ローカルデータメモリと、

２つの隣接する演算処理装置間の共用データ記憶装置とを含み、

該線形アレイの演算処理装置は各々、該ラウンドのうち１つを実現しかつ、その結果を連続する演算処理装置に転送し、それにより、該演算処理装置の線形アレイは演算処理装置パイプラインにおいて該暗号化アルゴリズムの連続的なラウンドを処理する、電子暗号化デバイス。

【請求項２４】　該暗号化アルゴリズムの実現中、該パイプライン内の各演算処理装置は、結果として得られるデータを、後続の演算処理装置によって直接

アクセスすることができるように該後続の演算処理装置と共用されるデータメモリ内に書込む、請求項２３に記載の電子暗号化デバイス。

【請求項２５】　演算処理装置の線形アレイを含む暗号化データ処理システムであって、各演算処理装置は、

命令メモリと、

該命令メモリからの命令を処理するためのプロセッサと、

データメモリとを含み、

該線形アレイの該演算処理装置のデータメモリは、該線形アレイの隣接する演算処理装置間でデータを転送するための、隣接する演算処理装置間で共用されるデュアルポートメモリを含む、暗号化データ処理システム。

【請求項２６】　各プロセッサは制御ユニットおよびＡＬＵを含み、該制御ユニット、ＡＬＵ、命令メモリ、および該演算処理装置のデータメモリは、ローカル演算処理装置データバスに接続され、該ローカルデータバスはスイッチによって２つの独立した区分に区分けされ、該制御ユニットおよび命令メモリは該区分のうち一方に接続されかつ、該ＡＬＵならびにローカルおよび共用データメモリは該区分のうち他方に接続される、請求項２５に記載の電子暗号化システム。

【請求項２７】　各演算処理装置は、該演算処理装置内で乗算演算を行なうための乗算器をさらに含む、請求項２５に記載の電子暗号化システム。

【請求項２８】　複数の演算処理装置の該乗算器は、幅のより広い乗算器の区分として連結されるように適合される、請求項２７に記載の電子暗号化システム。

【請求項２９】　各乗算器は部分積加算器を含み、該加算器は、独立した乗算器として動作しているときには第１の入力の組を選択し、また、連結されているときには隣接する演算処理装置からの入力を含む第２の入力の組を選択するための、入力選択回路を有する、請求項２８に記載の電子暗号化システム。

【請求項３０】　グローバルランダムアクセスメモリおよびグローバルバスをさらに含み、データは該グローバルランダムアクセスメモリと該演算処理装置データメモリとの間で該グローバルバスを通じて転送される、請求項２５に記載の電子暗号化システム。

【請求項３１】　該グローバルバスに結合されて、該演算処理装置によって処理されるデータ語よりも幅の広いデータ語を処理するための、中央処理装置をさらに含む、請求項３０に記載の電子暗号化システム。

【請求項３２】　該演算処理装置内で乗算演算を行なうための乗算器をさらに含む、請求項３１に記載の電子暗号化システム。

【請求項３３】　複数の演算処理装置の該乗算器は、より幅の広い乗算器の区分として連結されるように適合される、請求項３２に記載の電子暗号化システム。

【請求項３４】　各乗算器は部分積加算器を含み、該加算器は、独立した乗算器として動作しているときには第１の入力の組を、また、連結されているときには隣接した演算処理装置からの入力を含む第２の入力の組を選択するための、入力選択回路を有する、請求項３３に記載の電子暗号化システム。

【請求項３５】　該中央処理装置は加算器を含み、該加算器は、

複数加算器区分を含み、該複数加算器区分の各々はキャリ出力および合計出力を有し、該複数加算器区分の各々は２つのオペランドのうち各オペランドの１区分を処理し、該加算器はさらに、

加算器サイクル内でキャリが得られる限り、連続するクロックサイクル中、該キャリ出力を連続する加算器区分へのキャリ入力として選択するキャリ選択器と、

加算器サイクル内でキャリが得られる限り、連続するクロックサイクル中、各合計出力を同じ加算器区分へのオペランド入力として選択するオペランド選択器とを含む、請求項３１に記載の電子暗号化システム。

【請求項３６】　各演算処理装置の各プロセッサは、ＭｍｏｄＮを計算するモジュロ調整演算を行なう、請求項２５に記載の電子暗号化システム。

【請求項３７】　各演算処理装置の各プロセッサは、Ａ±ＢｍｏｄＮを計算するモジュロ加算または減算演算を行なう、請求項２５に記載の電子暗号化システム。

【請求項３８】　各演算処理装置の各プロセッサは、Ａ×ＢｍｏｄＮを計算するモジュロ乗算演算を行なう、請求項２５に記載の電子暗号化システム。

【請求項３９】　該暗号化デバイスは加算器をさらに含み、該加算器は、

複数の加算器区分を含み、該複数の加算器区分の各々はキャリ出力および合計出力を有し、該複数の加算器区分は２つのオペランドのうち各オペランドの１区分を処理し、該加算器はさらに、

加算器サイクル内でキャリが得られる限り、連続するクロックサイクル中、該キャリ出力を連続する加算器区分へのキャリ入力として選択する、キャリ選択器と、

加算器サイクル内でキャリが得られる限り、連続するクロックサイクル中、各合計出力を同じ加算器区分へのオペランド入力として選択する、オペランド選択器とを含む、請求項２５に記載の電子暗号化システム。

【請求項４０】　該暗号化アルゴリズムの実現中、該パイプライン内の各演算処理装置は、結果として得られたデータを、後続の演算処理装置が直接アクセスすることができるように該後続の演算処理装置と共用するデータメモリに書込む、請求項２５に記載の電子暗号化デバイス。

【請求項４１】　乗算器回路であって、該回路は、

複数の乗算器区分を含み、その各々が第１の長さのオペランド語を受取り、さらに、

該乗算器区分が別個の乗算器として動作しているときには第１の入力の組を選択し、また、第２の語長のオペランドに対する演算を行なう幅のより広い乗算器として該乗算器区分を連結するためには第２の入力の組を選択する、入力選択器を含む、乗算器回路。

【請求項４２】　各乗算器区分は部分積加算器を含む、請求項４１に記載の乗算器。

【請求項４３】　加算器であって、該加算器は、

複数の加算器区分を含み、その各々がキャリ出力および合計出力を有し、該加算器区分の各々は２つのオペランドのうち各オペランドの１区分を処理し、さらに、

加算器サイクル中にキャリが得られる限り、連続するクロックサイクル中、該キャリ出力を連続する加算器区分へのキャリ入力として選択する、キャリ選択器

と、

加算器サイクル中にキャリが得られる限り、連続するクロックサイクル中、各キャリ出力を同じ加算器区分へのオペランド入力として選択する、オペランド選択器とを含む、加算器。

【請求項４４】　電子暗号化デバイスであって、該デバイスは単一チップ上に、

演算処理装置の線形アレイを含み、その各々は、命令ストアと、データ記憶装置と、該命令ストアからの命令のシーケンスを処理して第１の長さのデータ語に対する演算を行なうプロセッサとを有し、該演算処理装置の該データ記憶装置は、該アレイの隣接する演算処理装置の間でデータを転送するために隣接する演算処理装置間で共用されるデュアルポートメモリを有し、該線形アレイの該演算処理装置は、自身の命令ストア内に、暗号化アルゴリズムのそれぞれのラウンドを記憶しかつ、該ラウンドの結果を連続する演算処理装置に転送し、よって、該演算処理装置の線形アレイは、演算処理装置パイプラインにおいて該暗号化アルゴリズムの連続するラウンドを処理し、さらに、

グローバルランダムアクセスメモリと、

該グローバルランダムアクセスメモリと該演算処理装置データメモリとの間でそれを介してデータが転送される、グローバルバスと、

少なくとも該第１の長さよりも長い第２の長さのデータ語に対する演算を行なう、公開鍵暗号化プロセッサとを含み、該公開鍵暗号化プロセッサは、該第２の長さの語長でグローバルランダムアクセスメモリにアクセスする、電子暗号化デバイス。

【請求項４５】　単一チップ上に演算処理装置のアレイを含む、電子暗号化デバイスであって、各演算処理装置は、

暗号化アルゴリズムのラウンドを記憶するための命令メモリ手段と、

該命令メモリからの該ラウンドを実現するためのプロセッサ手段と、

暗号化データオペランドおよび該ラウンドを実現することによって得られる暗号化されたデータを記憶するためのデータ記憶手段とを含む、電子暗号化デバイス。

【請求項４６】　該データ記憶手段は、その一部が、該線形アレイの隣接する演算処理装置の間でデータを転送するために該線形アレイの隣接する演算処理装置の間で共用される、請求項４５に記載の電子暗号化デバイス。

【請求項４７】　グローバルランダムアクセス手段およびグローバルバス手段をさらに含み、該グローバルランダムアクセス手段と該演算処理装置データ記憶手段との間のデータの転送は該グローバルバス手段を介して行なわれる、請求項４６に記載の電子暗号化デバイス。

【請求項４８】　該グローバルバス手段に結合されて、該演算処理装置によって処理されるデータ語よりも幅の広いデータ語を処理するための、中央処理手段をさらに含む、請求項４７に記載の電子暗号化デバイス。

【請求項４９】　暗号化方法であって、該方法は、

単一チップ上の電子回路において、暗号化されるべきデータを受取るステップと、

該データを該チップ上のデータ演算処理装置のパイプラインに与えるステップとを含み、各演算処理装置は、暗号化のラウンドを処理し、その結果を連続する演算処理装置に転送し、それにより、該演算処理装置が演算処理装置のパイプラインにおいて該暗号化アルゴリズムの連続するラウンドを実現する、方法。

【請求項５０】　結果は共用メモリを介して連続する演算処理装置に転送される、請求項４９に記載の方法。

【請求項５１】　該チップ上の、グローバルバスを介して該演算処理装置に結合された中央処理装置で、暗号化アルゴリズムを処理するステップをさらに含み、該中央処理装置は、該演算処理装置によって処理されるデータ語よりも幅の広いデータ語を処理する、請求項５０に記載の方法。

【請求項５２】　該チップ上の、グローバルバスを介して該演算処理装置に結合された中央処理装置において、暗号化アルゴリズムを処理するステップをさらに含み、該中央処理装置は、該演算処理装置によって処理されるデータ語よりも幅の広いデータ語を処理する、請求項４９に記載の方法。

【発明の詳細な説明】

【０００１】

【発明の分野】

本発明は高性能ネットワーク暗号化デバイスに関し、より特定的には、ハードウェアおよびソフトウェアの双方を組み込む暗号化デバイスに関する。

【０００２】

【発明の背景】

インターネットの出現以前は、企業のデータネットワークは典型的に、公共の電話会社からリースした専用遠隔通信ラインで構成されていた。このようなデータネットワークのハードウェア実装は、媒体上で絶対的な独占権を有する規制された公益企業であるその電話会社の独占的所有物であったために、セキュリティは大した問題ではなかった。その単一のプロバイダは、契約により安全が義務づけられ、また、中継ネットワークへの外部からのアクセスが不可能であったために、外部からのハッキングやタンパリングに対してかなり耐性があった。

【０００３】

今日、ますます多くの企業がインターネットに価値を見出している。インターネットは現時点において、単一のコンピュータネットワークとしては世界中で最も広く展開しているネットワークであり、したがって、国際的な企業ネットワークに容易に利用することが可能である。インターネットはまた、消費者レベルの製品であるので、インターネットアクセスは通常、専用電話会社のネットワークによって提供される同じサービスよりもはるかに低コストで提供され得る。また、インターネットのエンドユーザに対する可用性によって、個人が家庭や他の遠隔地から企業のネットワークに容易にアクセスすることが可能となっている。

【０００４】

しかしながら、インターネットは複数の株式会社によって運営されており、オープンプロトコルを用い、自由に調査可能なインバンドルーティングおよび制御の下に置かれている。このような環境は、ハッカーを育てるには絶好の土壌である。企業の諜報活動は今日では利益の多いビジネスであって、インターネット上でビジネスを行なう企業にとって、予防を怠ることは重大な損失を被ることにつ

ながる。

【０００５】

今日、インターネット上では、プライバシーおよび強力な認証のためのいくつかの基準が存在する。プライバシーは暗号化／復号化、すなわち解読、によって達成される。典型的に、暗号化／解読はメッセージ内容のプライバシーを維持しながらも当事者間でオープンチャネルを介してデータを転送することを可能にするよう設計されたアルゴリズムに基づいて行なわれる。これは、送信者が暗号化鍵を使用してデータを暗号化し、受信者が解読鍵を使用してそれを解読することによって達成される（ときに、暗号化鍵と解読鍵とは同一のものである）。

【０００６】

暗号化アルゴリズムの種類

暗号化アルゴリズムは、公開鍵アルゴリズムと秘密鍵アルゴリズムとに分類することができる。秘密鍵アルゴリズムにおいては両方の鍵が秘密であり、これに対し、公開鍵アルゴリズムでは鍵のうち一方が公開されている。ブロック型暗号は、今日使用されている秘密鍵暗号システムの代表である。通常、ブロック型暗号については、暗号化鍵と解読鍵とは同じものである。ブロック型暗号は、データのブロック、典型的には３２～１２８ビットを入力として受取り、同じ数のビットを出力として生成する。暗号化および解読は、長さが５６ビットから１２８ビットの間である鍵を使用して行なわれる。この暗号化アルゴリズムは、鍵を知らなければメッセージの解読が非常に困難なものとなるように設計されている。

【０００７】

インターネットのセキュリティプロトコルにより、ブロック型暗号に加えて、公開鍵アルゴリズムが多用されている。PogueおよびRivestに発行された米国特許番号第５，１４４，６６７号に記載の、Rivest, Shamir, Adelman（ＲＳＡ）暗号システム等の公開鍵暗号システムは、２つの鍵を使用し、そのうち一方のみが公開されている。ある人物が鍵を公開すると、他の誰もがその鍵を使用して秘密のメッセージをその人物に対して送信することができるが、そのメッセージは秘密鍵を使用しなければ解読することができない。このような公開鍵暗号化の利点は、会話を行なう前にすべての相手に対して秘密鍵を配布する必要がないこと

である。これに対して、もし秘密鍵の暗号化のみが使用されるようであれば、メッセージを受信する予定の各相手先に対して１つずつ、合せて多数の秘密鍵を生成せねばならず、またそれらすべての秘密鍵を１つずつ個別に配布せねばならなくなる。秘密裡に秘密鍵を送信しようと試みる場合、秘密鍵暗号化のみを使用してメッセージそのものを送信する場合と同じ問題が生じることになる。このような問題を、鍵の配布問題（key distribution problem）と呼ぶ。

【０００８】

鍵の交換は、公開鍵技術の別の応用例である。鍵交換プロトコルにおいて、当事者間の会話が第三者によって傍受された場合にも、当事者間は秘密鍵で対処することが可能である。米国特許番号第４，２００，７７０号に記載されているDiffie-Hellman指数関数鍵交換は、そのようなプロトコルの一例である。

【０００９】

ＲＳＡやDiffie-Hellman指数関数鍵交換等の、大半の公開鍵アルゴリズムは、モジュラ指数関数に基づいており、この関数は、α^{x}ｍｏｄＰを計算するものである。この式は、「αをｘ乗し、その解をｐで除して、剰余を得る」ことを意味する。このような計算を行なうには非常にコストがかかるが、それは以下の理由による。すなわち、この演算を行なうために、多数の乗算および除算を繰返す必要がある。ただし、１９８５年４月の、Mathematics of Computation, Vol. 44, No. 170の「試行除算を行なわないモジュラ乗算（“Modular Multiplication Without Trial Division”）」に記載のモンゴメリーの法（Montgomery's method）等の技術によれば、必要とされる除算の数を減じることができる。加えて、使用される数も非常に大きく（典型的に１０２４ビット以上）、したがって、一般のＣＰＵに見られる乗除命令を直接使用することができず、代わりに、そのような大きな乗算および除算を、１つのＣＰＵで行なうのに十分小さい演算へと分割する、特別なアルゴリズムを使用せねばならない。また、そのようなアルゴリズムの実行時間は通常、関連する機械語の数の二乗に比例する。これらの要因により、大きな数の乗算はその演算が非常に遅いものとなる。たとえば、Pentium（登録商標）は、１回の３２×３２ビット乗算を１０クロックサイクルで行なうことができる。２０４８ビット数は６４個の３２ビット語で表わすことができる。

２０４８×２０４８ビット乗算には、６４×６４回の別個の３２×３２ビット乗算が必要であり、この乗算のためにこのPentium上では４０９６０クロックが必要となる。２０４８ビット指数を用いる指数関数は、通常の方法では最高で４０９６回の乗算を必要とし、これには約１億６７００万クロックサイクルが必要となる。このPentiumが１６６ＭＨＺで動作するとすると、この演算全体でおよそ１秒かかることになる。驚くべきことに、この例では除算にかかる時間は一切考慮されていないのである。明らかに、Pentium等の一般的なＣＰＵで鍵の生成および交換を行なうことはほとんど期待することはできない。

【００１０】

公開鍵アルゴリズムは、計算が非常に面倒なので、典型的にメッセージ全体を暗号化するのには使用されず、代わりに、プライベート鍵暗号システムがメッセージの転送に使用される。メッセージの暗号化に使用されるプライベート鍵はセッション鍵と呼ばれ、この鍵が無作為に選ばれて公開鍵を用いて暗号化される。暗号化されたセッション鍵は暗号化されたメッセージとともに相手先に送信される。相手先は、自身の秘密鍵を用いてセッション鍵を解読し、その時点で、そのセッション鍵を用いてメッセージを解読することができる。各通信には異なるセッション鍵が使用されるので、もし１つのセッション鍵が破壊されたとしても、読むことができるメッセージはそのセッション鍵を使用して暗号化された１つのメッセージのみである。この公開鍵／プライベート鍵の方法はまた、双方向端末セッション等の、通常動作時には決して終了することのない連続的な通信を保護するのにも使用することが可能である。この場合、セッション鍵は、公開鍵生成技術を繰返すことによって周期的に（たとえば１時間ごとに）変更される。やはり、セッション鍵を頻繁に変更することによって、暗号化が破られたとしても犠牲となるデータ量は制限される。

【００１１】

先行技術

ソフトウェアをベースとした解決法を用いて企業のネットワークへのアクセスを可能にする、ネットワークレベルの暗号化デバイスが広く使用されている。Raptor Eagle Remote等の製品は、暗号化をすべてソフトウェアで行なっている。

ソフトウェアは、暗号器のスループットを制限する。公開鍵技術を用いたセッション鍵の生成には数分かかることもある。この理由のために、セッション鍵の再生成はある人々が望むほどには頻繁には行なわれない。しかし、ソフトウェアは、その分野における開発に応じて、暗号化アルゴリズムを容易に変更することができるという利点を有する。

【００１２】

他のデバイスは、ハードウェアとソフトウェアとの組合せを使用する。たとえば、Northern Telecom（現在のEntrust）のSentinel X.25暗号化製品は、ＡＭＤによって製造されたＤＥＳチップを使用して、ＤＥＳ秘密鍵の暗号化を行なう。ＤＥＳはハードウェアで効率的に実装されるように設計されたものなので、ハードウェア実装の方がはるかに高速である。ソフトウェアにおいては多くのＣＰＵ命令を要する転換を、並列の専用ルックアップテーブルおよび配線を使用して行なうことが可能である。

【００１３】

Sentinelはまた、Motorola DSP56000プロセッサを使用して、公開鍵演算を行なう。当時、ＤＳＰの単一サイクルの乗算能力のおかげで、この方法によって、一般的なＣＩＳＣマイクロプロセッサで公開鍵アルゴリズムを実現するよりもはるかに高速で演算ができるようになった。

【００１４】

大半のハードウェア暗号化デバイスにおいては、それによって実現することのできるアルゴリズムの数が大幅に制限されている。たとえば、Sentinelにおいて使用されるＡＭＤチップは、ＤＥＳのみを実行する。Ｈｉ／Ｆｎによるより最近のデバイスでは、ＤＥＳおよびＲＣ４を実行することができる。しかし、ＲＣ５またはＩＤＥＡを実現したい場合には、別の製品を用いねばならないであろう。

【００１５】

【発明の概要】

好ましい高性能プログラマブルネットワーク暗号化デバイスは、単一チップ内に集積され、これは、その命令の組が共通の暗号化アルゴリズムに対して最適化された、並列パイプライン式のプロセッサシステムである。本発明は、ハードウ

ェアおよびソフトウェアの両方の方法の利点を実現する。該プロセッサはプログラマブルプロセッサであるため、どのような暗号化アルゴリズムも実現することが可能であり、これは、１つのアルゴリズムのみを実行するよう設計されるハードウェア実装の暗号化プロセッサとは対照的である。しかし、このプロセッサのアーキテクチャは、暗号化に有益な特性である並列計算を可能にしているので、その性能は、専用ハードウェアデバイスの性能により近づく。

【００１６】

本発明の好ましい実現例に従えば、電子暗号化デバイスは演算処理装置のアレイを含む。各演算処理装置は、暗号化アルゴリズムの１ラウンドを記憶するための命令メモリを含み、該ラウンドは、命令の１シーケンスを含む。該演算処理装置はまた、命令メモリからのラウンドを実現するためのプロセッサ、ならびに、暗号化データオペランドおよびラウンドを実現することによって得られる暗号化されたデータを記憶するためのデータ記憶装置を含む。該アレイの各演算処理装置は、複数のラウンドのうち１つを実現し、その結果を連続する演算処理装置へと転送し、それにより、該演算処理装置のアレイは演算処理装置のパイプラインにおいて暗号化アルゴリズムの連続的なラウンドを実現する。

【００１７】

好ましい実施例においては、該データ記憶装置はその一部分が該線形アレイの隣接する演算処理装置間で共用されており、該線形アレイの隣接する演算処理装置間でデータを転送するのに使用される。該共用データ記憶装置は好ましくは、デュアルポートメモリで構成されるが、これはまた、共用レジスタを含んでもよい。

【００１８】

好ましい演算処理装置は、制御ユニットおよびＡＬＵを含む。制御ユニット、ＡＬＵ、命令メモリおよびデータ記憶装置は、ローカルデータメモリおよび共用データメモリも含めて、ローカル演算処理装置バスに接続される。このローカルバスはスイッチによって区分されて、命令メモリおよび制御ユニットを接続するローカル命令バス区分と、ＡＬＵ、ローカルデータメモリおよび共用データメモリを接続するローカルデータバス区分とに分けられる。該スイッチは、該２つの

ローカルバス区分上で別個に同時に演算ができるようにするか、または、それら２つのバス区分の間で通信ができるようにする。各演算処理装置はさらに、その演算処理装置内で乗算演算を行なうための乗算器を含む。

【００１９】

好ましい暗号化デバイスはさらに、グローバルランダムアクセスメモリおよびグローバルバスを含み、データは該グローバルランダムアクセスメモリと演算処理装置のデータ記憶装置との間で該グローバルバスを通じて転送される。中央処理装置は、このグローバルバスに結合されて、演算処理装置によって処理されるデータ語よりも幅の広いデータ語を処理する。複数の演算処理装置のそれぞれの乗算器は、中央処理装置によって使用されるより幅の広い乗算器の区分として連結できるようにされ得る。好ましくは、各乗算器は部分積加算器を含み、該加算器は、個別の乗算器として動作しているときには第１の入力の組を選択し、かつ、連結されているときには隣接する演算処理装置からの入力を含む第２の入力の組を選択するための、入力選択回路を有する。

【００２０】

好ましくは、中央処理装置は新規な加算器を含む。該加算器において、複数加算器区分の各々はキャリ出力および合計出力を有し、それら加算器区分の各々は、２つあるオペランドの各オペランドの１区分を処理する。選択器は、加算器サイクル中にキャリが得られる限り、連続するクロックサイクル中、該キャリ出力を連続する加算器区分へのキャリ入力として選択する。選択器はまた、各合計出力を同じ加算器区分へのオペランド入力として選択する。したがって、加算器サイクル中にキャリが得られる限り、ある加算器の合計出力はその入力にフィードバックされ、また該加算器区分は先行するサイクルにおいて先行する区分からキャリ出力として生成されたキャリ入力を受取ることになる。

【００２１】

好ましくは、各演算処理装置は、除算回路を用いずにＭｍｏｄＮを計算する、モジュラ調整演算を行なう。各演算処理装置はまた、Ａ±ＢｍｏｄＮを計算するモジュロ加算／減算演算を行なう。さらに、各演算処理装置は、Ａ×ＢｍｏｄＮを計算するモジュロ乗算演算を行なう。

【００２２】

【詳細な説明】

本発明の上記および他の目的、特徴および利点は、添付の図面に示される本発明の好ましい実施例に関する以下のより詳細な説明から明らかとなるであろう。添付の図面においては、複数の図面を通じて、同じ部分には同様の参照符号が付される。図面は必ずしも一定の比で描かれているわけではなく、本発明の原理を説明するために強調されている部分を含む。

【００２３】

本発明の暗号化チップは、任意のアプリケーションにおける１または複数のデータストリーム上で、共通のデータ暗号化および復号化、すなわち解読、のアルゴリズムを行なうようプログラムすることが可能である。この暗号化チップの主要な目的は、インターネット上でその使用が想定されるアルゴリズムを用いて、１００～２０００Ｍｂｐｓのデータレートで、高速データ暗号化を行なうことである。

【００２４】

アプリケーションの例を図１Ａおよび図１Ｂに示す。図１Ａにおいて、ソース２２からのデータが、暗号化チップ２４で暗号化された後に公衆ネットワーク２６に渡される。データはその後、暗号化チップ２８内で解読されて、宛先３０に送られる。一実施例においては、このソースおよび宛先自体が、ローカルエリアネットワーク等のネットワークである。そのような場合、これら暗号化チップが、ローカルエリアネットワークと公衆ネットワーク２６との間に安全な経路を提供する。

【００２５】

図１Ｂに示されるリンク暗号化アプリケーションにおいては、各リンク内でルータ間で転送されるデータが暗号化される。この場合、リンクとリンクの間にあるルータ３２に入力された暗号化データは、暗号化チップ３４でまず解読されねばならず、またそのデータは、暗号化チップ３６において、次のリンクの暗号化アルゴリズムに従って再度暗号化される。

【００２６】

今日、ＤＥＳ、ＲＣ５およびＩＤＥＡという３つの主要な秘密鍵ブロック型暗号化アルゴリズムが一般に使用されている。最初の２つのアルゴリズムは、標準的なインターネットプロトコルセキュリティ（Internet Protocol SECurity）の略である、ＩＰＳＥＣ標準アルゴリズムである。ＩＤＥＡは、広く利用されている電子メール暗号化プログラムであるＰＧＰによって使用されるアルゴリズムである。

【００２７】

典型的に、ブロック型アルゴリズムは、多数のラウンドで構成され、各ラウンドは、暗号化アルゴリズムにおける演算の１シーケンスである。暗号化アルゴリズムを完全に実現するには、８～３２ラウンドが必要とされる。各ラウンドによって行なわれる演算は、しばしば同じものであるが、同じものでなくてもよい。ソフトウェアにおいては、各ラウンドは少数の機械命令で実現される。ハードウェアにおいては、各ラウンドは専用回路で実現される。ハードウェアは典型的にパイプライン化されており、各ラウンドは自身に該当するパイプライン段において実現される。

【００２８】

図２は、本発明の一実施例に従った、集積チップの解決法を図示する。これを今後、暗号化チップと呼ぶ。暗号化チップと呼ぶと、そのチップが暗号化を行なうことができることが示唆されるが、このチップが復号化、すなわち解読、およびメッセージダイジェスト機能もまた行なうことに留意されたい。

【００２９】

データは、ネットワークデータを受取る入力段４０を介して、典型的にはシリアルビットストリームとして暗号化チップに入力される。イーサネット、ＡＴＭまたは他のどのような直列化フォーマットも使用することができる。入力段はこのシリアルデータストリームを、暗号化／解読パイプラインへの入力として処理するのに好適な、ブロック整合されたデータへと変換する。入力ブロックのサイズはプログラム可能である。図２に示した好ましい実施例においては、パイプラインは線形アレイに配された複数の演算処理装置３７からなり、各演算処理装置は、命令メモリ、レジスタファイル、ＡＬＵ、ローカルおよび共用データメモリ

、ならびに制御回路を含む。演算処理装置の各々は、３２ビット幅のデータ語を処理するよう設計されている。暗号化されたデータは、該パイプラインの最後の演算処理装置から取出されて出力段４２に渡され、出力段４２がそのブロックデータをシリアルストリームフォーマットに戻して、そのデータをネットワークを介してまたは局所宛先へと送る。

【００３０】

データは、グローバルデータバス３８を介して、暗号化チップ内の隣接しない演算処理装置間および／または他の装置間で転送することができる。グローバルデータバス３８にはまた、Ｉ／Ｏ通信ロジック５４が接続されており、このロジック５４が、ホストＣＰＵ（図示せず）との通信を可能にする。ホストＣＰＵとの通信は、暗号化チップを使用前にプログラムするのに必要である。グローバルランダムアクセスメモリ（ＲＡＭ）４４もまたグローバルデータバス３８に接続され、それにより、演算処理装置間でグローバルな通信が可能となっている。制御ＣＰＵ５２は、暗号化パイプラインプロセッサの動作を同期化する。このＣＰＵは、ＭＩＰＳ、ＡＲＭまたはＡＲＣ等の、利用可能ないずれの組込み型ＣＰＵコアを使用しても実現することができる。さらに、公開鍵暗号化アルゴリズムのように非常に幅の広いオペランドを利用するアルゴリズムを処理することができるように、公開鍵（ＰＫ）コアプロセッサ４６が制御ＣＰＵ５２に接続されている。ＰＫコアは、８個から１６個の５１２ビット幅のレジスタからなるレジスタファイル４８、およびＰＫ　ＡＬＵ５０を含む。ＰＫコアプロセッサは、１システムクロックサイクルで、５１２ビットバスを介してグローバルＲＡＭ４４との間でデータの送受信を行なうことができる。５１２ビットのオペランドは、典型的には２～３２クロックサイクルで、ＡＬＵ５０内で処理される。ＰＫコアＡＬＵ５０は、制御ＣＰＵ５２によって制御されるコプロセッサであって、ローディングおよび記憶の他には、算術および論理演算のみを行なう。ＰＫアルゴリズムを実現するのに必要な他の命令は、制御ＣＰＵ５２内で実行され得る。

【００３１】

この暗号化チップは、秘密鍵アルゴリズムの各ラウンドのためのコードを、パイプラインの別個の演算処理装置内で実現する。計算が終わると、１つのＰＥか

らのデータは次のＰＥに転送され、そこで次のラウンドが実現される。第１のＰＥはその後、入来するデータの次のブロックのための暗号化ラウンドを処理することができるようになる。パイプライン処理は残りのＰＥにおいて続けられる。このアーキテクチャを用いて１つのブロックを暗号化するのに必要とされる時間は、したがって、１つのラウンドを暗号化するのに必要とされる時間に等しい。

【００３２】

多くのブロックアルゴリズムは、データを暗号化するのにある演算の組を使用し、鍵を拡張するのに別の演算の組を使用する。鍵の拡張は、比較的小さい鍵（５６～１２８ビット）を、統計的に無作為の性質を有するより大きい数（５１２ビット以上）の鍵へと変換するプロセスである。こうして拡張された鍵は、より小さなサブ鍵に分配され、拡張された鍵の異なる部分が各々異なるラウンドのために使用される。拡張された鍵がデータによって変化しないことに注目することが重要である。したがって、これはクリティカルパス内にはないため、予め計算してメモリに記憶しておくことができる。後に説明するコードの例は、鍵情報が予め計算されて各ＰＥのローカルデータメモリ内に記憶されているものと仮定している。

【００３３】

ブロックアルゴリズムの基本的なアプリケーションは、平文（暗号化されていない情報）のブロックを同じサイズの暗号文（暗号化された情報）のブロックに変換したり、その逆を行なう。この動作モードは、電子コードブック（ＥＣＢ）モードとして知られているが、これはセキュリティに関して多くの固有の弱点を有するので、基本的な出力のいくつかを入力に戻るよう巡回させることによって暗号化にフィードバックを導入する方法が一般に使用されている。この暗号化チップは、グローバルデータバス３８を利用して暗号フィードバック（ＣＦＢ）を行なう。ＥＣＢモードにおいては、データの新しいブロックを各パイプラインサイクルにつき１回暗号化することができる。これは１０～１００個の命令であり得る。しかし、ＣＦＢモードにおいては、各データはパイプラインを多数回通過せねばならない。このモードは単一チャネル上のスループットを大幅に減じるが、パイプラインにおいてインターリーブされている多数のデータチャネルを暗号

化することによって、ピーク性能を達成することができる。

【００３４】

本発明の一実施例に従った１つの演算処理装置ＰＥのブロック図を図３に示す。演算処理装置３７は、８～１６個の３２ビットレジスタで構成されたレジスタファイル５８から得られる３２ビット語の演算を行なう、ＡＬＵ５６を含む。レジスタファイル５８およびＡＬＵ５６は、制御ユニット６０によって制御される。制御ユニット６０は、演算処理装置命令メモリ６２からの命令をデコードする。各演算処理装置命令メモリは暗号化アルゴリズムの少なくとも１つのラウンドを記憶し、ここで１つのラウンドとは、暗号化アルゴリズムにおける命令の１シーケンスと定義される。各演算処理装置がアクセスすることのできるＰＥデータメモリスペースは、４つの領域に分割される。すなわち、ローカルＰＥメモリ６４（図３においてはPE_nローカルメモリ）、共用メモリ６６（図３では、ｎ番目の演算処理装置とｎ－１番目の演算処理装置との間で共用される、$PE_{n,n-1}$共用メモリ）、第２の共用メモリ６８（図３では、ｎ＋１番目の演算処理装置とｎ番目の演算処理装置との間で共用される、$PE_{n+1,n}$共用メモリ）、および、図２を参照して説明した、すべてのＰＥがアクセス可能なグローバルメモリ４４、の４つの領域である。これらのメモリはすべて、１つの演算処理装置、たとえばｎ番目の演算処理装置のアドレススペースにマップされる。どの種類のメモリにアクセスするのにも、特別な命令は必要ない。すべてのメモリはすべてのメモリアクセス命令によってアクセス可能である。

【００３５】

１つの演算処理装置のメモリ６６および６８は、デュアルポートＳＲＡＭであって、これらはそれぞれ、先行する、すなわち前隣りのパイプ段および、次の、すなわち後ろ隣りのパイプ段と共用される。あるＰＥにとっての後ろ隣りのＰＥとの共用メモリは、次のＰＥにとっての前隣りのＰＥとの共用メモリと同じものであることを理解されたい。

【００３６】

これらのデュアルポートのＳＲＡＭは、パイプライン段を通じてデータを伝搬するのに使用される。ある演算処理装置が、転送されるべきデータをそれに関連

する後ろ隣りの装置との共用メモリに書込む。すると、その記憶されたデータを、該当する後ろ隣りの演算処理装置が、自身の前隣りの装置との共用メモリから読出す。ここで、前隣りの装置との共用メモリとは、上述のように、先行する演算処理装置にとっての後ろ隣りの装置との共用メモリと同一無二のメモリを指す。これらのメモリはデュアルポートメモリであるため、アクセスにはタイミングの制限がない。アクセスの同期化は、ソフトウェアの作者または編集者による機械命令の静的なスケジューリングを用いて行なわれる。さらに、隣接するＰＥ間の通信にグローバルバスを使用しないので、ＰＥはすべて同時に通信することが可能である。

【００３７】

グローバルメモリ４４はグローバル通信バスに接続される。任意の時間にグローバルメモリ４４にアクセスが許可されるのは１つの演算処理装置のみである。このメモリは、たとえば、フィードバック暗号化アルゴリズム中に、隣接していない演算処理装置間でデータをやりとりするのに使用され、また、個々の演算処理装置のための補助記憶装置としての役割を果たす。

【００３８】

ＰＥ命令メモリ６２は、現代のＲＩＳＣプロセッサの整数ユニットのそれに似た、命令の組を有する。この命令の組は、どのレジスタもどの命令に対するオペランドとしても使用することができるという点で、いくぶん直交性である。浮動小数点やメモリ管理サポートは、どちらも暗号化には有益ではないので、設ける必要はない。しかし、この命令の組は、以下の有益な追加機能を含む。すなわち、モジュラ加算／減算命令、モジュラ乗算命令およびモジュロ調整命令、である。

【００３９】

モジュラ加算／減算命令は、Ａ±ＢｍｏｄＮを計算する（「ＭｍｏｄＮ」の数はＭをＮで除した際の剰余である）。図１５Ａから図１５Ｄは、モジュラ加算、減算および調整を、１つのスリーインワン（3-in-1）モジュロ算術ユニットに組合せた例を示す。

【００４０】

図１５Ａは、モジュラ加算演算を示す。加算すべき２つの数ＡおよびＢが双方ともＮよりも小さければ、加算器１２０からのそれらの合計を、Ｎを法として減じることができる。すなわち具体的には、減算器１２２においてＮを減じ、その後、その差の符号に応じて、マルチプレクサ１２４を介して、減算器の出力または元々の数のいずれかを選択する。同様に、図１５Ｂに示すモジュラ減算演算の場合には、２つの数ＡおよびＢがＮよりも小さい場合には、Ｎを法とするそれらの差を計算することが可能である。これは具体的には、減算器１２８からの差が負であれば加算器１２６においてＮを加算し、その差が正であればマルチプレクサ１３０を介してその差を選択することによって、行なわれる。ここで、モジュロ加算およびモジュロ減算がいずれも除算を必要としないことに注目されたい。しかし、それらは、連続２回の加算を必要とする（そのうち１つは合計／差を計算するもの、もう１つはＮを法として減算するものである）。このような２回続けての加算がクリティカルパスに打撃を与える場合には、Ｎを法とする減算は、別個の命令としてエンコードすることが可能であり、これを「モジュロ調整」命令と呼ぶ。

【００４１】

図１５Ｃに示すこのモジュロ調整命令は、ＡおよびＢの両方が既にＮを法として減じられていて、ＭがＡとＢとの合計または差のいずれかであるものとして、ＭｍｏｄＮを計算する。Ｍが負である場合、ロジック１３２は、加算器／減算器１３４においてＭにＮを加えて、マルチプレクサ１３６を介して結果が生成されるようにする。Ｍが正である場合には、ロジック１３２は、Ｎの減算を行ない、その差が正であればその差を返し、その差が負であればＭを返す。この命令は、合計および差の命令と関連づけて使用することが可能であり、それにより、モジュラ加算／減算命令が不要となる。

【００４２】

図１５Ｄにおいて、スリーインワンの算術ユニットは、モジュロ加算、モジュロ減算およびモジュロ調整を、各演算処理装置内で実現される単一のユニットに組合せる。１つの命令（モジュラ加算、減算または調整）および最上位ビット（ＭＳＢ）の符号入力に応答するロジック１４４の制御下で、加算器／減算器１３

８は装置１２０および１２８のいずれかの機能を行ない、加算器／減算器１４０は、装置１２２、１２６および１３４のうちいずれかの機能を行なう。マルチプレクサ１４２は装置１２４、１３０および１３６に対応する。モジュロ調整演算において、ＭがＡ入力に印加され、Ｂ入力はゼロにセットされる。この組合せユニットは、速度は落ちるが、面積効率は最も高い。この組合せユニットはまた、Mathematics of Computation, Vol. 44, No. 170, April 1985, pages 519-521の、ピーター・Ｌ・モンゴメリー（Peter L. Montgomery）による「試行除算を行なわないモジュラ乗算」に記された、試行除算を行なわないモジュラ乗算のためのモンゴメリーの法を実現するのに有益である。

【００４３】

モジュラ加算および減算は、従来技術によるプロセッサにおいてわずか２～３個の命令で実現することができるが、これらの命令を暗号化チップの命令の組の特別な関数として含むことで、特定的な暗号化アルゴリズムの場合においては、わずかながら高速化につながる。

【００４４】

モジュラ乗算命令は、Ａ＊ＢｍｏｄＮを計算する。この命令に使用される乗算器は、下により詳細に説明する。暗号化チップは、後に明らかとなるであろう理由によって、全体のモジュラ乗算命令を提供することができる。

【００４５】

表１は、以下の例において使用される、ＰＥの命令の組の代表的な例を示す。他の従来技術によるＲＩＳＣ命令もまた実現することが可能である。

【００４６】

【表１】

表１:命令の組のサンプル

命令	記述
load rn, addr store rn, addr	レジスタｎにメモリをロードする レジスタｎをメモリに記憶する
xor r1, r2, r3 add r1, r2, r3 rol r1, r2, r3	r1 = r2 xor r3 r1 = r2 + r3 r1 = r2 <<< r3(<<<は、回転命令のためのJava演算子である。32ビットのオペランドに対して、ビット31がビット位置0に回転される。)
xor r1, addr add r1, addr rol r1, addr	r1 = r1 xor メモリ[addr]r1 = r1 + メモリ[addr] r1 = r1 <<< メモリ[addr]
moda r1, r2 moda r1, addr	モジュロ調整：r1 = r1 mod r2。ここで、r1はモジュラ加算または乗算の結果である。 r1 = r1 mod メモリ[addr
mul r1, r2, r3 mulm r1, r2, r3, r4	乗算：r1 = r2×r3。32ビットで実行される。 モジュラ乗算：r1 = r2×r3 mod r4
Jump label sync label	制御を無条件にラベルに転送する。 パイプライン同期化：すべてのPEが「sync」命令に到着するまで待ち、その後ラベルに分岐する。
Dbra rn, label	rn = rn - 1；もしrn !=0であれば、ラベルにジャンプする。
cbra r1 cond r2, label	比較および分岐：r1とr2を比較し、条件が真であればラベルに分岐する。「Cond」は、==, !=, <, >, <=または>=のうちの１つである。

【００４７】

レイアウトの課題

暗号化チップの一般的なレイアウトを図４に示す。ここでは、１６個の演算処理装置および、５１２ビット幅の公開鍵ＰＫコアユニットを想定する。ここで５１２ビットのＰＫコア語幅を選択したのは、そのレイアウトが容易であるためである。たとえば１０２４ビット幅は、より広いシリコン面積を必要とするであろうが、性能は倍加するであろう。

【００４８】

個々の素子は、図２および図３に示した素子に匹敵し得る。１６個の演算処理装置が、レイアウトの大きな領域内で左下側に１列に線形に配されており、その１つが詳細に示されている。図中、共用乗算器素子７０は、図示された演算処理装置に関連づけて示されている。前述のように、３２×３２乗算器区分７０は、

各演算処理装置と関連づけられて、それぞれの演算処理装置内で３２ビット乗算を行なう。これに代えて、乗算器素子７０は、公開鍵ＡＬＵ５０のための幅の広い５１２×３２ビット乗算器として機能するように、連結することも可能である。公開鍵ＰＫ　ＡＬＵ５０は、秘密鍵ＳＫ素子の右側に配置され、上述のような演算処理装置で構成されている。ＰＫ　ＡＬＵの隣りに、ＰＫレジスタファイル４８が配される。ＰＫ　ＡＬＵ５０およびＰＫレジスタファイル４８は併せて、図２において４６で示されたＰＫ処理コアを形成する。ＰＫコアの右側には、グローバルメモリ（ＲＡＭ）４４が配置される。チップの上辺に沿って、制御ＣＰＵ５２、通信ロジック５４および入出力処理ブロック４０、４２が配置される。グローバルデータバス３８は、ＳＫ素子、ＰＫコア４６、グローバルＲＡＭ４４、通信ロジック５４および制御ＣＰＵ５２を繋ぐ。

【００４９】

ローカルバス接続を含む典型的な演算処理装置のレイアウトを図５に示す。１つの演算処理装置のすべての構成要素は、ローカル演算処理装置データバス７２を介して通信することができる。このバス７２は、メモリとレジスタとの間のすべての転送を扱う。ここで、次に隣接するＰＥとの共用ＰＥメモリ６８は、図示されている演算処理装置の他の素子と直列に（in-link）配されており、これに対し、先に隣接するＰＥとの共用ＰＥメモリ６６は、先に隣接する演算処理装置の素子と直列に配されていることに注目されたい。プログラミングおよびテストの目的のために、すべてのＰＥメモリはグローバルバス３８からアクセス可能である。スイッチ７４は通常、ローカルバス７２をグローバルバス３８から切離しているが、ローカルＲＡＭ６４とグローバルＲＡＭ４４との間でデータ転送を可能にするように選択的に閉じることが可能である。別のスイッチ７６は、ローカルバス７２を独立した２つの区分に区分けすることを可能にし、これにより、制御ユニット６０は、バス７２上のデータ転送と同時に、ＲＡＭ６２から命令を読出すことが可能となる。このように、演算処理装置内の動作は、ある命令がＰＥ

ＡＬＵ５６内で実行されている間に次の命令が制御ユニット内で処理されるというように、パイプライン化することが可能である。暗号化コードの実行中、スイッチ７４および７６は通常は開かれており、これにより、命令ＲＡＭからの命

令フェッチを、データメモリおよびレジスタファイルからのデータフェッチと同時に進めることができる。

【００５０】

多数のマルチプロセッサアーキテクチャが提案されているが、それらの大半は、汎用マルチプロセッシングのために設計されている。このため、演算処理装置間の通信は通常、あるＰＥから別のＰＥへとデータを切換えるよう動的に構成することが可能な、切換マトリックスを使用して行なわれる。これらのスイッチの設計は非常に複雑である。このようなスイッチは暗号化には不要であるため、本発明の実施例においては、切換回路が大幅に減じられた、より簡単なＰＥの線形配列を用いている。

【００５１】

加えて、相互配線技術として、文献に記載されているようなＩ／Ｏポートを使用するのではなく共用メモリを使用していることにより、はるかに簡単かつはるかに強力なプログラミングモデルが生成される。ここで、２つのＰＥ、ＡおよびＢが単一の３２ビットＩ／Ｏポートに接続されているものとする。ＡがＢに対してデータの複数語を転送するためには、Ａは各語をＩ／Ｏポートに書込んで、Ｂがそれを読出すのを待たねばならない。これに対し、ＡおよびＢが、通信のすべての語を保持するのに十分な大きさの共用メモリによって接続されている場合には、ＡはＢが読出すのを待つことなくそのデータを書出すことが可能である。さらに、ＰＥ　Ｂはどのような順序でもそれらの語を読出すことができ、また、そのデータから、進行中のジョブに応じて適宜、必要なものをピックアップして選択することもできる。最後に、共用メモリのうちあるメモリが通信に不要である場合には、そのようなメモリはローカルメモリの延長として使用されて、付加的なローカルワークスペースを提供することができることに注目されたい。

【００５２】

公開鍵サポート

効率的な公開鍵暗号化のためには、公開鍵コプロセッサによって提供される効率的なモジュラ指数関数が必要である。このユニットは、以下の項目を含む。すなわち：

・１６個の５１２ビット幅のレジスタで構成される、ＰＫレジスタファイル４８
・連結されたＳＫ乗算器素子からなる、ＰＫ５１２×３２ビット乗算器７０（このユニットは、わずか３２クロックサイクルで１つの５１２×５１２乗算を行なうことができる）
・ＰＫ５１２ビット加算器ＡＬＵ５０、これは、２～１６サイクルで、典型的には２サイクル以下で、加算を行なうことができる
・単一クロックサイクルで５１２ビット語をロードおよび記憶するために、ＰＫコプロセッサからの５１２ビット並列アクセスのために構成される、グローバルメモリ４４。

【００５３】

ＰＫコアプロセッサは、モジュラ乗算を５１２ビット語を使用して行なうことによって加速する。本発明のＰＫユニットを用いる５１２×５１２乗算演算は、下に説明する１６個の演算処理装置を連結した乗算器素子を用いて、１６個の５１２×３２乗算を行なうことによって実現されるであろう。各乗算につき２クロックサイクルが必要とされかつそのような乗算が１６回必要とされると仮定すると、１回の５１２×５１２乗算に必要なのは３２クロックサイクルであり、１回の２０４８×２０４８乗算はわずか５１２クロックサイクルで行なうことができることになる。４０９６回の乗算を必要とする全体のモジュラ指数演算は、合計２００万クロックサイクルを要することになるが、これは、先に説明したPentiumの例に比べて８０倍の改良を意味する。ＰＫアルゴリズムにおいても同様の性能の改良が期待される。これは、先行技術に比べて大幅な性能の向上を意味し、セッション鍵をより頻繁に変更することが可能になって、セキュリティが向上することを意味する。

【００５４】

５１２ビット加算器

加算器は、公開鍵ＰＫユニットと秘密鍵ＳＫユニットとの間で共用されることはない。加算演算および論理演算がＰＫおよびＳＫの双方において共通であるので、各ユニットは自身の加算器を有し、したがって、演算を同時に進行することが可能である。

【００５５】

公開鍵ＰＫ　ＡＬＵ５０内において、５１２ビットの単一サイクルの加算器は非常に複雑であって、ＡＬＵのクリティカルパス時間を大幅に増やすことになるであろう。このため、ＡＬＵ５０内の５１２ビット加算器は、図６に示すように、１６個の３２ビット加算器から形成される。動作中、ＡＮＤゲート７８およびマルチプレクサ８０がまず、２つの３２ビットオペランド区分を、３２ビット加算器Ａ０～Ａ１５の各々に供給する。ここで、ＡＮＤゲート７８は３２ビット幅の動作を表わす。各３２ビット加算器は、キャリ出力に加えて、３２ビット合計を計算する。１つの加算器のキャリ出力は、Ｄフリップフロップ７９を介して、次の加算器のキャリ入力に接続される。第１のサイクル中にキャリが生成されると、それはフリップフロップ内にクロック入力され、そのフリップフロップにおいてそのキャリは、次のクロックサイクルのためのキャリ入力として利用可能となる。各合計は、Ｄフリップフロップ８１およびマルチプレクサ８０を介して同じ加算器の一方入力に戻される。この加算器の他方入力は、連続するクロックサイクル中、ＡＮＤゲート７８を用いてゼロに保持される。合計を各加算器へのキャリ入力として戻し加算するステップは、３２ビット加算器のうちいずれかの出力にキャリが得られる限り繰返される。

【００５６】

５１２ビット加算器の動作は、以下の例を参照してよりよく理解されるであろう。この例においては、実際の実装時の１６個の３２ビット語の代わりに、４つの４ビットの２進語を使用する。

【００５７】

【数１】

加算：	1101	0110	1001	1011	
	0001	0101	1100	1011	
	01110	01011	10101	10110	キャリ出力は
					0,0,1,1 である。
	1110	1011	0101	0110	
	0	1	1	0	先のキャリ
	1110	1100	0110	0110	最終合計

【００５８】

ここでは、さらなるキャリがもはや得られない最終合計に達するまでに必要とされる加算は２回であった。これは典型的な場合である。加算器が暗号化演算のために使用されるので、加算される回数はある程度ランダムにばらつくと仮定すると安全であろう。初回の加算の後にキャリ出力が得られる可能性は極めて高い。しかし、最下位ビットとして戻され加算されるキャリによって最上位ビットからの別のキャリが得られる可能性は極めて低い。このため、ほとんどの加算演算はわずか２クロックサイクルしか必要としないと予測されるのである。

【００５９】

５１２ビット加算器を構築するという最初の課題に戻って、標準的なキャリ先見型またはキャリバイパス型加算器設計を使用する場合、その加算器を通じるクリティカルパスは極めて長くなるであろう。なぜなら、キャリが、５１２ビットの演算を行なう何らかの最適化された回路を通じて伝搬されねばならないためである。この加算器は極めて大きくかつ低速であろう。これに対して、本発明の一実施例においては、５１２ビット加算器は３２ビット加算器から構成されている。３２ビット加算器の設計は今日ではよく知られておりまた十分に最適化されている。個々の３２ビット加算器の最大クロック速度は、５１２ビットキャリ先見型設計のクロック速度の２倍以上であると予測される。したがって、本発明に従った２以上のサイクルの加算器は、より大きな５１２ビット加算器よりも、チップ面積の消費量はより少ないのに対し、通常はより高速で動作することができるであろう。

【００６０】

最悪の場合、下に説明するように、１６個の３２ビット加算器の実装において、キャリを有さない最終合計を完全に計算するのに、１６サイクルが必要となることも考えられる。ここで再び４ビットの２進語の例を使用して説明すると、以下のようになる。

【００６１】

【数２】

加算：1111 1111 1111 1111
0000 0000 0000 0001
01111 01111 01111 10000 第１のキャリ出力は０,０,０,１である。

1111 1111 1111 0000
0 0 1 0
01111 01111 10000 00000 第２のキャリ出力は０,０,１,０である。

1111 1111 0000 0000
0 1 0 0
01111 10000 00000 00000 第３のキャリ出力は０,１,０,０である。

1111 0000 0000 0000
1 0 0 0
10000 0000 0000 0000

【００６２】

以上のように、４回の加算が必要であった。一般に、ｎ個の数のグループについては、最大でｎ回の加算が必要とされる。

【００６３】

５１２×３２乗算器

乗算器は占有面積が広い。各秘密鍵演算処理装置は、たとえば下により詳細に説明するＩＤＥＡ等の、乗算を必要と秘密鍵アルゴリズムを実現するためには、自身の乗算器を含まねばならない。各ＰＥ乗算器によって占められる面積を合わせると相当な面積となり、そこで、この面積は、５１２×３２ビット公開鍵乗算器を実現するのに使用される。面積の節約のために、このように大きな５１２×３２乗算器は、各秘密鍵演算処理装置において１６個の３２×３２乗算器を連結することによって実現される。換言すれば、秘密鍵ユニットおよび公開鍵ユニットは、図４のチップレイアウト内に示すように、複数の乗算器素子を共用することが可能である。したがって、乗算器素子の使用は、秘密鍵演算処理装置とＰＫコアプロセッサとの間で調整されねばならない。なぜなら、ＰＫコアプロセッサは、複数の秘密鍵演算処理装置のうちどの１つが独立して乗算演算を行なってい

る場合にも、乗算演算を行なうことができないためである。

【００６４】

乗算器の連結を説明するために、４×４／４×Ｎ乗算器の組合せの簡単な設計を下に示す。ただし、Ｂｏｏｔｈの符号化および４：２コンプレッサ等の、より進歩した乗算器設計技術もまた利用可能である。以下に、簡単な実現例を提示する。

【００６５】

【数３】

```
      1 0 1 1
    x 0 1 0 0
      0 0 0 0
    1 0 1 1          部分積
  1 0 1 1
0 0 0 0
1 0 0 0 0 1 0
```

【００６６】

１桁の乗算は、ＡＮＤゲートを用いることによって容易に実現することができる。２つの４ビットオペランドを使用した場合、その結果は、部分積の１６ビットから構成される。これらの部分積は、効率的に加算されねばならない。部分積はたとえば、２つの４ビット全加算器および１つの６ビット全加算器を使用して加算することができるが、それらは部分積の加算を行なうのに相当な時間を要するであろう。なぜなら、キャリを複数の加算器を通じて伝搬させる必要があるためである。このような加算器実装の全体としての結果は、遅すぎるであろう。よりよい方法として、加算器のキャリが通らねばならない段の数がより少なくて済むような加算器が考えられる。

【００６７】

好ましい乗算器の基本的な構成要素は、３つの入力をとってそれら入力の２ビットの合計を出力する、全加算器である。図７に全加算器を、符号を用いて示すが、ここでは、２進数の代わりに四角形を使用して、一般化および簡素化を図っている。上方の３つの四角は全加算器の３つの入力を示し、下方の２つの四角は

合計出力およびキャリ出力を示す。キャリが左下側にあるのは、その桁の値が合計のそれの２倍であることを示すためである。

【００６８】

４×４乗算器の加算の第１の段を図８に示す。合計線の上方にある１６個の四角は、そのいくつかが黒で示され、その他は白い箱として示されているが、これらは、加算されねばならないある部分積のビットを表わしている。黒で示されるビットは、この第１の段において、４つの全加算器８２を使用して加算されるものである。白い箱で示すビットは、第１段では加算されずに、図８に矢印で示すように、次の加算段に備えて単に下方に送られるビットである。第１の段における加算器の合計は、合計線の下に示されている。

【００６９】

第２の段を図９に示す。矢印はやはり、この現時点の段においては演算されずに単に下に送られるビットを示し、黒い箱で示されるビットは、この現時点における（すなわち第２の）段において加算されるべきビットを示す。ここでもやはり、黒い箱で示したエレメントが４つの全加算器８４を使用して加算される。全加算器８４によって生成される第２の段の出力には２つの数があり、これらは今度は一般的な４ビットキャリ加算器８６で加算されねばならない。

【００７０】

さまざまな加算器および乗算器アーキテクチャの性能を比較することは、本発明に従った乗算器の利点を説明するのに役立つであろう。４ビット加算器の簡単な実現例は、図１２に示すように、直列に並んだ４つの全加算器Ａ０～Ａ３で構成される。この設計においては、最も右側の加算器のキャリ出力C_{out}が、その左側にあるすべての加算器段のそれぞれに影響を及ぼす可能性がある。この設計におけるクリティカルパスはしたがって、４加算器段である。典型的な全加算器が２以上の論理段から構成されるので、１つの４ビット加算器の合計ゲート遅延は８段を超える場合がある。

【００７１】

改良された４ビット加算器は、キャリ先見型設計のものである。３ビットのキャリ先見型加算器を図１３に示す。４ビット設計はこれよりもわずかに複雑であ

る。ＡＮＤゲート１０２、ＯＲゲート１０４および排他的ＯＲゲート１０６の動作の詳細な説明は、周知の回路であるためここでは省略する。キャリ先見型加算器の利点は、キャリが最終合計ビットまでわずか４論理ゲートで伝搬することである。より大きな数に対するより複雑な設計は、より多くの論理段を有するが、それでもキャリ連鎖型設計よりは、やはり高速である。

【００７２】

全４×４乗算器において、キャリ保存型設計は、２つの全加算器および最後のキャリ先見型加算器を通じてクリティカルパスを作る。全加算器のみを使用した実現例では、クリティカルパスがより長くなるであろう。なぜなら、連鎖型キャリを使用する簡単な加算器は、キャリ先見型加算器よりも低速だからである。最後に、部分積合計の最初の２つの段で全キャリ先見型加算器を使用した場合には、結果として得られる乗算器はやはり低速となるであろう。なぜなら、キャリ先見型加算器は個々の全加算器よりは低速なためである。なお、本発明に従った乗算器設計は、同じ部分積レベルでは、あるキャリをある加算器から別の加算器へと伝搬することはない。このようにして、乗算器を通じるクリティカルパスが、部分積合計の最初の２段において、２つを超える数の全加算器を含むことを確実に防止している。

【００７３】

図１０は、はるかに幅の広い４×Ｎ乗算器を示す。大きな黒い箱８２、８４、８６は、図９において使用されていたのと同じ全加算器ハードウェアを示す。この場合、全加算器が必要であるが、これは、各状況において３つの入力が合わせて加算されるためである。図９においては、すべての状況において３つの入力の加算が必要とされたわけではなかったので、より簡単な回路を使用することができた。しかし、４×Ｎを処理することのできるシステムを作るためには、すべての段において好ましくは全加算器が使用され、加えて、２つ以下の入力の場合にどのように処理を行なうべきかを決定する何らかの付加的な回路が必要となる。したがって、デュアルモードの加算器が複数個作られ、そのいくつかは１つの乗算器を有し、この乗算器が自身の複数入力のうち１つを供給することで、先行する段の出力または単一ビットの部分積の、どちらかが選択されるようにする。

【００７４】

図１１は、図１０に示す囲んだ領域８２、８４、８６を実現するのに必要とされる全加算器Ａを示し、合せてその左下方に、それぞれのキャリ出力を示す。好ましい実現例においては、各加算器Ａは全加算器である。加算器のうちいくつかは、４×４の場合（すなわち秘密鍵の場合）２つの入力のみを有し、これに対し、他の加算器は、４×Ｎの場合（すなわち公開鍵の場合）３入力を有する。２入力の加算器は、その第３の入力がイネーブル信号でゲート制御されるようにされねばならない。いくつかの加算器はまた、複数入力のうち１つを提供して先の段の出力または単一ビットの部分積のいずれかを選択するようにする、乗算器を必要とする。下方に示されたキャリ先見型加算器８６は、４×４の場合に積の最終ビットを生成するために、４つの位置毎に１つのキャリ出力を必要とする。

【００７５】

図１１において、４×４乗算器の部分積は、以下の部分積のシナリオに対応するように参照符号が付されている。

【００７６】

【数４】

```
    A B C D
  E F G H
I J K L
M N O P
```

【００７７】

４×Ｎ乗算器については、隣接する部分積もまた考慮に入れねばならない。それらは図１１において、以下のシナリオに従って参照符号が付されている。

【００７８】

【数５】

```
    D'  A B C D
    E F G H
  I J K L I'
M N O P M'
```

【００７９】

ここで、D′は、隣接する（左側または右側の）Ｄの等価物である。８ビットの最終合計は、Ｓ７、Ｓ６、Ｓ５、Ｓ４、Ｓ３、Ｓ２、Ｓ１、Ｓ０で示され、左側に隣接する乗算器の合計の下方３ビットは、Ｓ２′、Ｓ１′、Ｓ０′で示される。２：１乗算器８８は、選択信号Ｓｅｌを有する。一般に、Ｓｅｌが論理１である場合、左側の入力がマルチプレクサの出力に渡され、反対にＳｅｌが論理０の場合には、右側の入力がマルチプレクサの出力に渡される。Ｓｅｌ信号はまた、ＡＮＤゲート９０をゲート制御するのにも使用される。Ｓｅｌが論理１である場合、ＡＮＤゲートへの他方入力が出力に渡され、反対に、Ｓｅｌが論理０である場合、ＡＮＤゲート９０はディスエーブルされて、他方入力の値にかかわらず論理０を渡す。したがって、図１１の実現例においては、Ｓｅｌが論理１である場合には４×Ｎ乗算器の区分が実現可能となり、積は出力Ｓ６～Ｓ３に現われる。Ｓｅｌが論理０であれば、４×４乗算器が実現されて、８ビットの積が出力Ｓ７～Ｓ０に現われる。このように、図１１の実現例は秘密鍵乗算器素子を示し、これは、１つの乗算器素子としても利用することができ、または、他の同様の乗算器素子と連結されて、はるかに幅の広い公開鍵乗算器を実現するのにも使用することができる。

【００８０】

実現例

先に説明した暗号化チップの好ましい実施例を参照して、一般的な暗号化アルゴリズムの実現例を以下に説明する。ＲＣ５はおそらくは、実現するのが最も簡単な暗号化アルゴリズムのうちの１つであろう。これは基本的に、３つの種類の演算を利用する。すなわち、ＸＯＲ、加算および回転である。これらすべては、表１に示すように、上述の演算処理装置のうちのいずれかによってサポート可能である。ＲＣ５は可変長のブロックを有するが、最も一般的には、ＲＣ５アルゴリズムの各ラウンドは、Ｓｉ１およびＳｉ２に記憶される６４ビットデータブロックおよび鍵値について演算を行なう。これらは、各演算処理装置内の定数であって、そのラウンドおよびその鍵のみに依存する。データを暗号化するために、６４ビットの入力ブロックは２つの３２ビット語に分割され、それらはその後、

前隣りのメモリ内の場所ＡおよびＢに記憶される。出力ブロックは、後ろ隣りのメモリにおけるＡ＿ｎｅｘｔおよびＢ＿ｎｅｘｔに書込まれることになる。ＲＣ５の暗号化アルゴリズムの１ラウンドの例を以下に示す。

【００８１】

【数６】

```
ループ：
    load r1, A
    xor r1, B
    rol r1, B
    add r1, Si1
    store r1, A_next
    load r2, B
    xor r2, r2, r1
    rol r2, r2, r1
    add r2, Si2
    store r2, B_next
    sync Loop
```

【００８２】

各ラウンドは１１クロックサイクルを必要とする。暗号化チップが最高４００ＭＨＺで動作し得る論理プロセスを用いるように設計されている場合には、１秒あたり３６００万ブロックを暗号化することが可能である。これは、ＥＣＢモードにおいて２８８ＭＢ／ｓに相当する。１２ラウンド（ＲＣ５における典型的な例）を想定すると、同じクロック速度で動作する従来技術によるＣＰＵと比較して、本発明の一実施例に従った複数ＰＥの同時実行によって、従来技術によるソフトウェア実装に対して１２倍も性能が改良されることになる。

【００８３】

ＩＤＥＡは、利用可能なブロックアルゴリズムのうち最も安全なものの１つであり、その構造ははるかにより複雑である。ＩＤＥＡは、６４ビットの平文ブロックに対して演算を行ない、１２８ビットの鍵が使用される。同じアルゴリズムを暗号化および解読の両方に使用する。このアルゴリズムの主要な原理は、種々の代数グループの演算、すなわち、ＸＯＲ、加算モジュロ2^{16}および乗算モジュロ$2^{16}+1$等の演算を組合せることである。これらの演算を使用して、１６ビットブロックに対する演算を行なう。

【００８４】

したがってＩＤＥＡは、モジュラ乗算およびモジュラ加算の両方を使用するが、それらはソフトウェアでは費用が高くつく演算である。乗算は、ＩＤＥＡのゼロの扱いによって複雑化されている。すなわち、乗算において、ゼロは（－１）モジュロ６５５３７と解釈されるのである。この値６５５３７が演算処理装置のレジスタファイルのレジスタｒ８内にプリロードされていると仮定し、また、レジスタｒ０がゼロを含むものと仮定して、以下に乗算マクロの例を示す。

【００８５】

【数７】

```
    MACRO MMULT(A,B,RESULT)
      cbra A = = r0, L1
      load RESULT, #1
      sub RESULT, B, RESULT
      jump L2
L1:
      cbra B = = r0, L3
      load RESULT, #1
      sub RESULT, A, RESULT
      jump L2
L3:
      mulm A, B, RESULT, r8
      andi #0xFFFF,RESULT
L2:
    ENDMACRO
```

【００８６】

ＩＤＥＡの各ラウンドは、モジュラ乗算、モジュラ加算および排他的ＯＲから構成される。１２８ビットの鍵がサブ鍵へと分割される。各演算処理装置のサブ鍵は、その鍵およびその演算処理装置のみに応じて変化するので、予め計算してＰＥ内に記憶させておくことが可能である。ＩＤＥＡに入力される平文は、先に説明したように、１６ビットの４つのサブブロックＸ１～Ｘ４から構成される。各ラウンドは、６つのサブ鍵Ｋ１～Ｋ６を使用し、以下のようにコード化することができる。

【００８７】

【数８】

```
ループ：
    load r1, X1
    load r9, K1
    MMULT r1, r9, r1
    load r2, X2
    load r9, K2
    MMULT r2, r9, r2
    load r3, X3
    load r9, K3
    MMULT r3, r9, r3
    load r4, X4
    load r9, K4
    MMULT r4, r9, r4
    xor r5, r1, r3
    xor r6, r2, r4
    load r9, K5
    MMULT r5, r9, r5
    add r5, r6
    and r6, #0xFFFF
    load r9, K6
    MMULT r6, r9, r6
    add r5, r6
    and r5, #0xFFFF
    xor r1, r6, r1
    xor r3, r6, r3
    xor r2, r5, r2
    xor r4, r3, r4
    store r1, X1_next
    store r2, X3_next
    store r3, X2_next
    store r4, X4_next
    sync Loop
```

【００８８】

ＩＤＥＡは８ラウンドを有するので、本発明の一実施例に従った暗号化チップハードウェア実装はその実行を８倍以上加速する。さらなる加速は、ほとんどのマイクロプロセッサにおいては利用されないモジュラ乗算命令によってもたらされる。上述のコードは、１ラウンドを実行するのにおよそ５０クロックサイクルを要する。４００ＭＨＺにおいて、この暗号化チップは、ＩＤＥＡで６４ＭＢ／ｓのレートで暗号化することができ、これは、チューリッヒのＥＴＨ大学（ETH University, Zurich）において開発された２５ＭＨＺハードウェア実装よりも約３倍高速である。

【００８９】

データ暗号化標準、すなわちＤＥＳは、当初、ハードウェア実装のために設計されたものであり、したがって、ソフトウェアで実現するのが最も困難なアルゴリズムである。それでも、本発明の一実施例に従えば、これは暗号化チップにおいて容易にコード化することが可能である。

【００９０】

先の２つのアルゴリズムと同様に、ＤＥＳもまた、６４ビットブロックでデータを暗号化するブロック型暗号である。平文の６４ビットブロックが入力であり、６４ビットの暗号文が出力となる。ここでもやはり、暗号化と解読の両方が同じアルゴリズムを使用し、ＤＥＳを対称的なアルゴリズムとしている。ＤＥＳは、この場合においては５６ビットの単一の鍵から、サブ鍵を作成する。これらのサブ鍵は、該当のＰＥおよびその５６ビットの鍵に応じて変化するものであり、したがって、それらは予め計算しておくことが可能である。

【００９１】

図１４に示すようなＤＥＳにおける基本的な概念は、鍵に基づいてテキスト上で代入を行ない引続き置換を行なうものである。以下の演算によってＤＥＳのコアが作られている。

・拡張：６４ビットブロックを２つの３２ビット片１０８、１１０に分割する。一方の片は暗号化によって影響を受けることがない。（これらの片は１つおきのラウンドで演算される。）影響を受ける方の片が８つの４ビットのグループに分割される。各グループは、それに隣接する２つのビットをコピーすることによって拡張される。

・拡張された各グループは、１１２においてサブ鍵でＸＯＲ処理される。

・ＸＯＲの６ビットの結果を使用して、Ｓボックス（S-box）と呼ばれる、６４エントリ×４ビット先見テーブル１１４をインデックスする。８個のグループの各々が自身のＳボックスを使用する。

・Ｓボックスからの出力は１１６において置換され、それらのビットがスクランブルされる。８個の出力から３２ビットが得られる。

・その３２ビットの出力が、１１８において、そのブロックの他方の３２ビット

片とＸＯＲ処理される。

【００９２】

これらの演算は以下のようにコード化することが可能である。すなわち：拡張は、入力語をコピーしてから、ビットをマスキングすることによって、１つが偶数のＳボックス入力を表わし他方が奇数のＳボックスの入力を表わす２つの語が存在するようにすることによって行なわれる。これら２つの語を、鍵情報でＸＯＲ処理し、その結果を使用して、Ｓボックスルックアップテーブルをインデックスする。各Ｓボックスにおけるデータは予め置換され、したがって、Ｓボックスの出力は３２ビットのデータとなる。最終値はすべての構成要素の論理ＯＲである。コードの例を以下に示す。

【００９３】

【数９】

```
ループ：
    load r1, A
    load r2, B
    load r3, r2
    store r2, A_next
    and r2, #0xF9F9F9F9
    and r3, #0x9F9F9F9F
    xor r2, K1
    xor r3, K2
    load r5, r0

    load r4, r3
    rol r4, #1
    and r4, #0x3f
    or r5,[r4 + S1]

    load r4, r2
    ror r4, #3
    and r4, #0x3f
    or r5,[r4 + S2]

    load r4, r3
    rol r4, #7
    and r4, #0x3f
    or r5,[r4 + S3]

    load r4, r2
    ror r4, #11
    and r4, #0x3f
    or r5,[r4 + S4]

    load r4, r3
    rol r4, #15
    and r4, #0x3f
    or r5,[r4 + S5]

    load r4, r2
    ror r4, #19
    and r4, #0x3f
    or r5,[r4 + S6]

    load r4, r3
    rol r4, #23
    and r4, #0x3f
    or r5,[r4 + S7]

    load r4, r2
    ror r4, #27
    and r4, #0x3f
    or r5,[r4 + S8]

    xor r5, r1, r5
    store r5, B_next
    sync Loop
```

【００９４】

このサンプルコードは、１ラウンドを実行するのに４４クロックサイクルを必要とする。４００ＭＨＺにおいて、７２ＭＢ／ｓのデータレートが達成され得る

。このレートは、１～３５ＭＢ／ｓの範囲のレートで暗号化を行なう、１９９０年代半ばに利用可能となったＤＥＳのハードウェア実装に比べて遜色のないものである。ＶＬＳＩテクノロジー（VLSI Technology）のＶＭ００７は、最高２００ＭＢ／ｓで暗号化を行なうことが可能である。

【００９５】

以上の例の各々において、その性能は、従来技術におけるＣＰＵ上のソフトウェア実装よりもはるかに高速であるが、専用ハードウェア実装よりも低速であることが示されている。本発明のハードウェア実装に対する利点は、暗号化チップがプログラマブルであり、したがって、今後想定され得るものも含むどのようなアルゴリズムも実装が可能であるということである。

【００９６】

特定的な公開鍵アルゴリズムの例は何ら示さなかったが、既存の方法に対して同様の改良が、本発明の好ましい実施例において説明したのと同様の技術を用いて実現され得るものと理解されたい。

【００９７】

均等物

本発明をその好ましい実施例を参照して特定的に図示しかつ説明したが、当業者においては、その形および詳細に、前掲の請求の範囲によって規定される本発明の精神および範囲から離れることなく、種々の変更が行なわれ得ることが理解されるであろう。当業者においては、日常的な作業の範囲を超えることなく、ここに特定的に示した本発明の具体的な実施例に対する多くの均等物が認識されるかまたは確認されるであろう。そのような均等物は、前述の請求の範囲に包含されるものと意図される。

【図面の簡単な説明】

【図１Ａ】　本発明の可能な応用例のブロック図である。

【図１Ｂ】　本発明の可能な応用例のブロック図である。

【図２】　本発明を用いた暗号化チップのブロック図である。

【図３】　図２の暗号化チップにおける演算処理装置のブロック図である。

【図４】　図２および図３に示した回路の好ましいチップレイアウトを示す

。

【図５】　図４に示したレイアウトに対応するように書き直された図３の演算処理装置ならびに、ＰＥローカルバスおよびグローバルバス接続を示す。

【図６】　図２のＰＫ　ＡＬＵにおいて使用される加算器回路を示す。

【図７】　ＰＫ　ＡＬＵの乗算器において使用される全加算器の符号を示す。

【図８】　全加算器を使用する４×４乗算器の第１段における処理を示す。

【図９】　４×４乗算器の３つの段を示す。

【図１０】　４×４乗算器の加算器がその上に重ねられた、幅の広い乗算器の加算器を示す。

【図１１】　図１０に示した広い語長の演算器において、同様の乗算器と連結されるように適合された、４×４乗算器のブロック図である。

【図１２】　全加算器を使用する８ビット加算器の従来技術による実現例を示す。

【図１３】　キャリ先見型加算器の従来技術による実現例である。

【図１４】　ＤＥＳ暗号化ラウンドをブロック図で表現したものである。

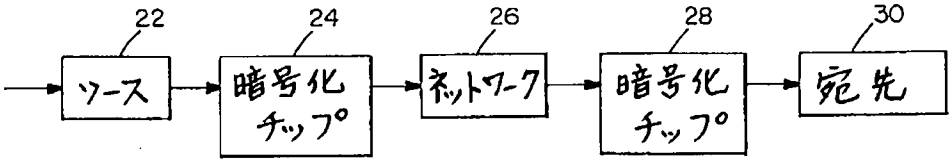
【図１５Ａ】　本発明の実施例に従ったモジュラ加算演算を示す機能図である。

【図１５Ｂ】　本発明の実施例に従ったモジュラ減算演算を示す機能図である。

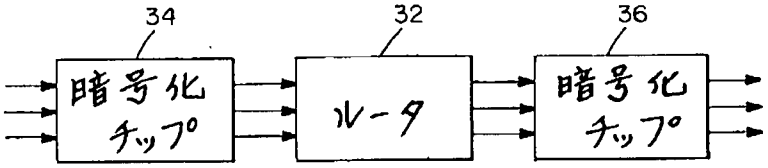
【図１５Ｃ】　本発明の実施例に従ったモジュラ調整演算を示す機能図である。

【図１５Ｄ】　本発明の実施例に従った３つすべてのモジュラ演算の組合せを示す機能図である。

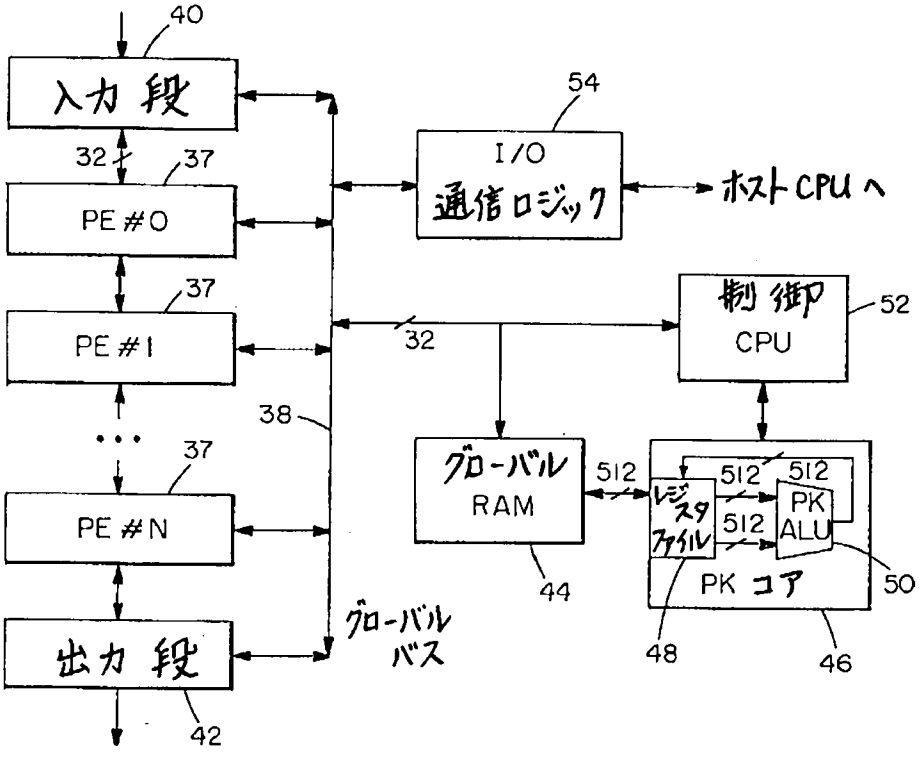
【図１Ａ】


22
24
26
28
30
ソース
暗号化チップ
ネットワーク
暗号化チップ
宛先

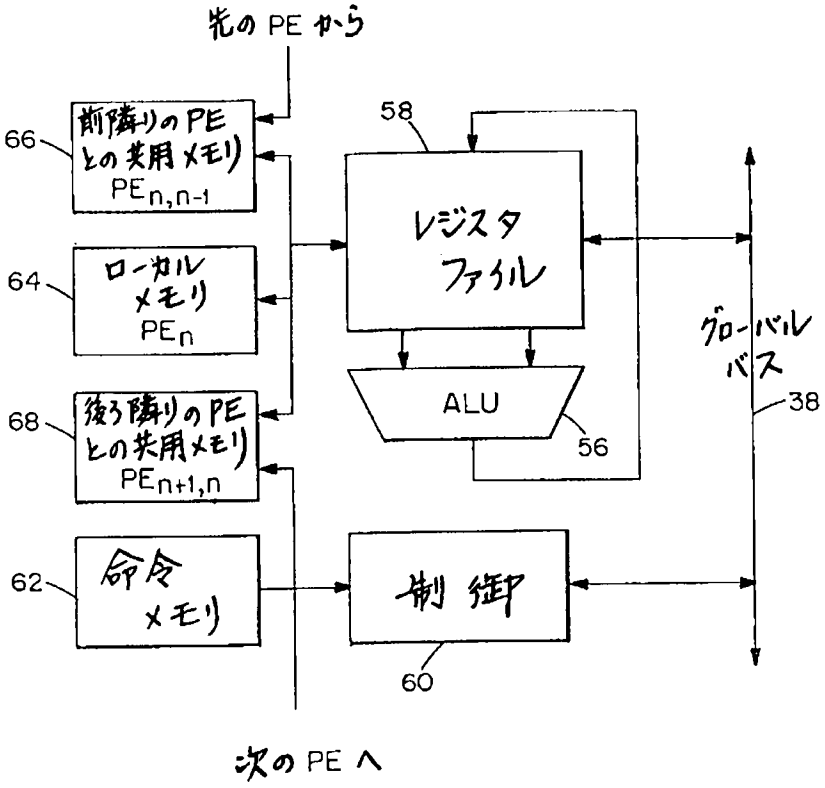
【図１Ｂ】


34
32
36
暗号化チップ
ルータ
暗号化チップ

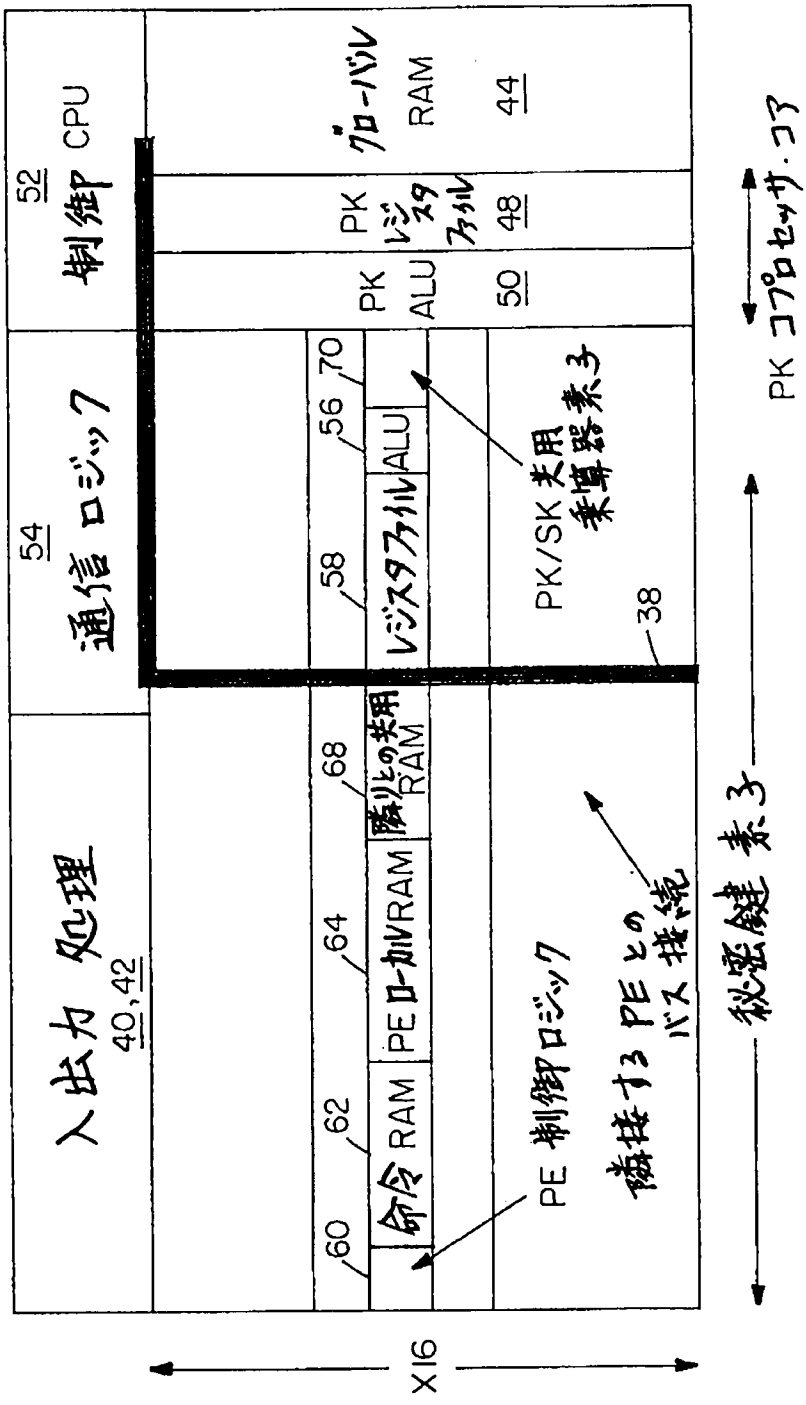
【図２】


40
入力段
32
37
PE #0
37
PE #1
37
PE #N
出力段
42
38
グローバルバス
54
I/O
通信ロジック
ホストCPUへ
32
制御
CPU
52
グローバル
RAM
44
512
512
512
512
レジスタファイル
PK
ALU
PK コア
48
46
50

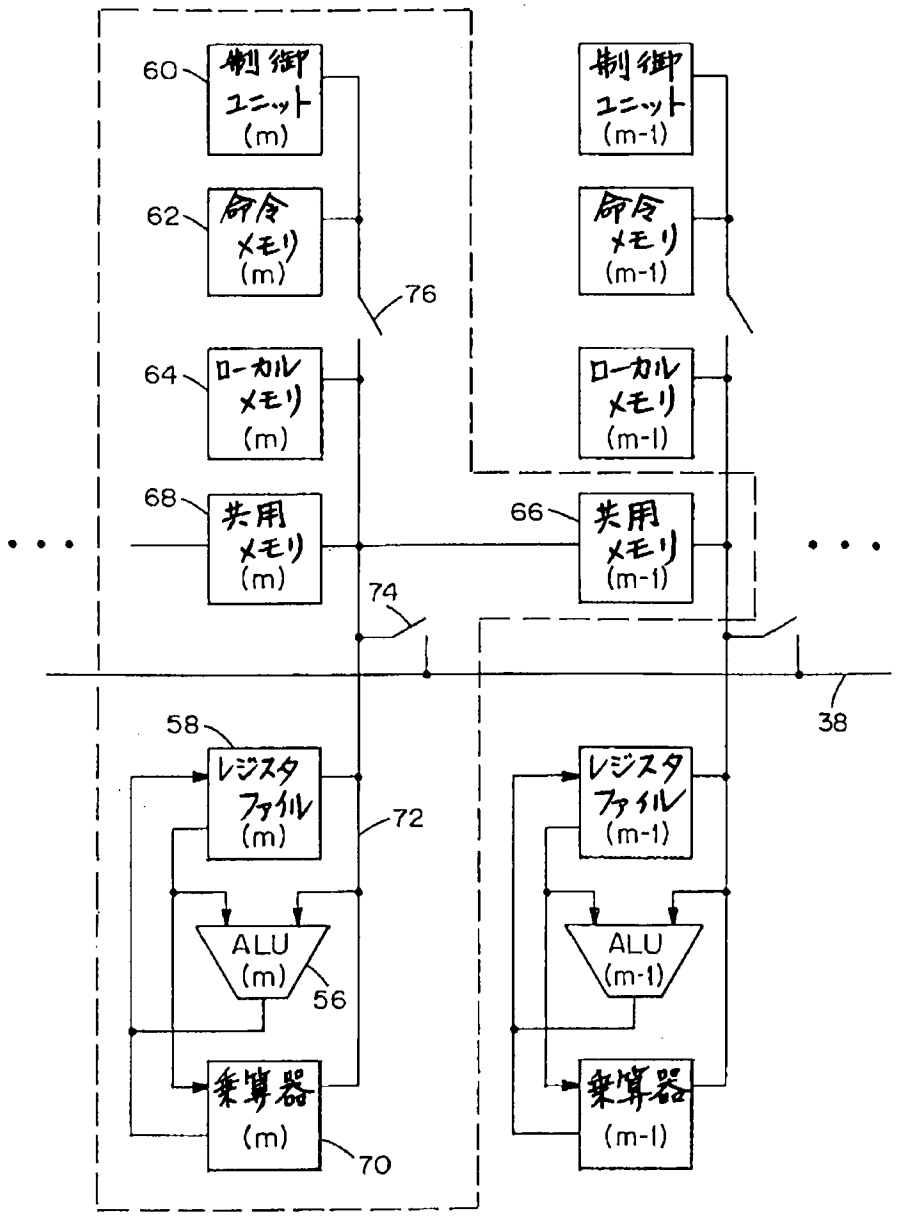
【図３】


先のPEから
前隣りのPEとの共用メモリ
PEn,n-1
66
ローカルメモリ
PEn
64
後ろ隣りのPEとの共用メモリ
PEn+1,n
68
命令メモリ
62
58
レジスタファイル
ALU
56
制御
60
グローバルバス
38
次のPEへ

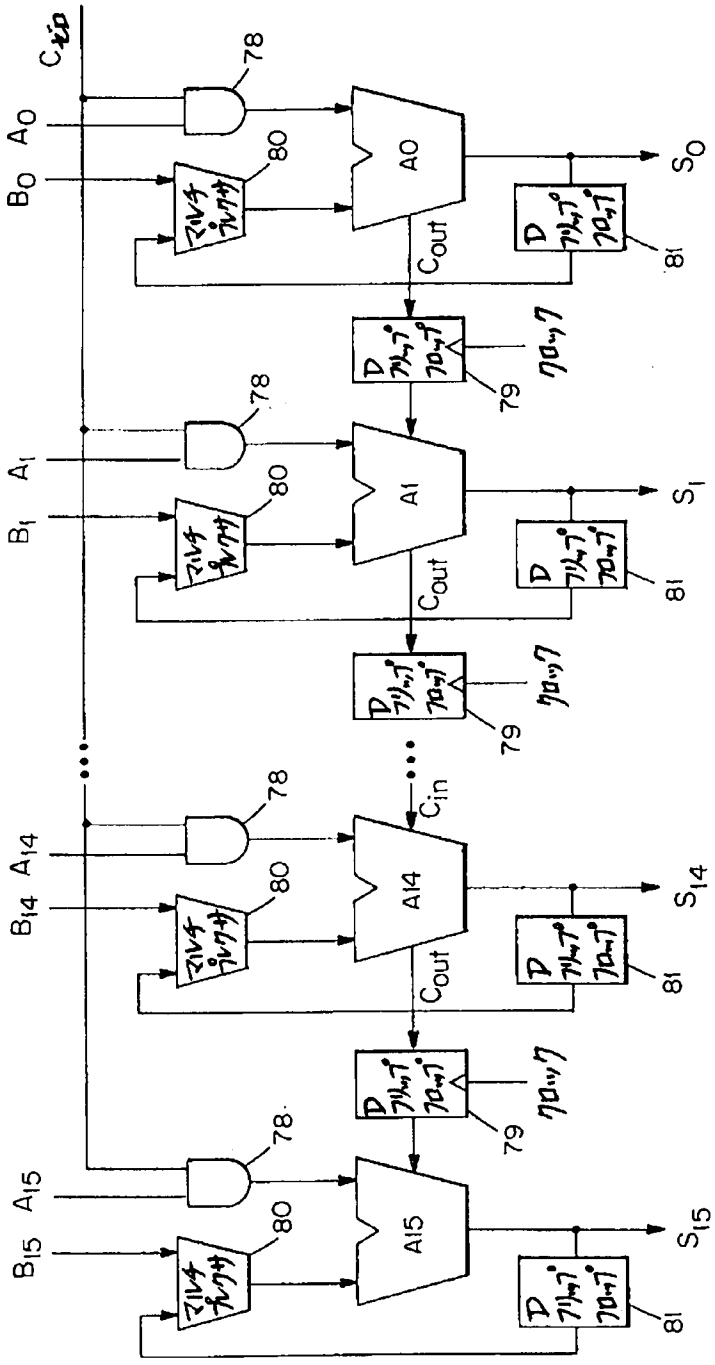
【図４】


入出力 処理
40,42
52
制御 CPU
54
通信 ロジック
グローバル RAM
44
PK レジスタファイル
48
PK ALU
50
PK/SK 共用 乗算器素子
PK コプロセッサ・コア
秘密鍵 素子
PE 制御ロジック
隣接する PE との バス接続
38
56
70
58
レジスタファイル
ALU
68
隣りとの共用 RAM
64
PE ローカル RAM
62
命令 RAM
60
X16

【図５】


60
制御ユニット (m)
62
命令メモリ (m)
64
ローカルメモリ (m)
68
共用メモリ (m)
76
74
58
レジスタファイル (m)
72
ALU (m)
56
乗算器 (m)
70
制御ユニット (m-1)
命令メモリ (m-1)
ローカルメモリ (m-1)
66
共用メモリ (m-1)
レジスタファイル (m-1)
ALU (m-1)
乗算器 (m-1)
38

【図６】


A0
B0
A1
B1
A14
B14
A15
B15
78
80
81
79
A0
A1
A14
A15
Cout
Cin
S0
S1
S14
S15
D
マルチプレクサ
フリップフロップ
クロック

【図７】

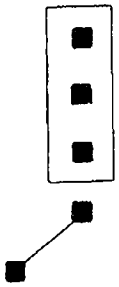


FIG. 7

【図８】

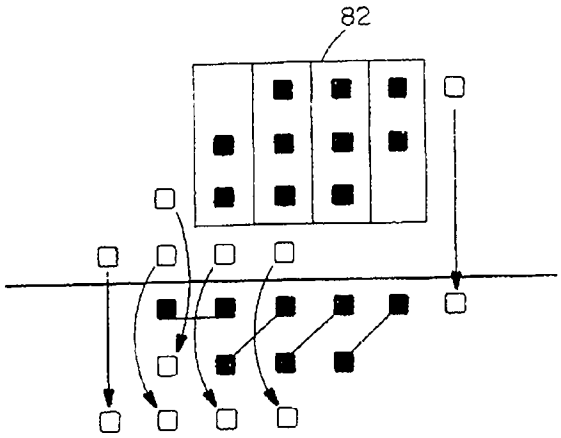


FIG. 8

【図９】

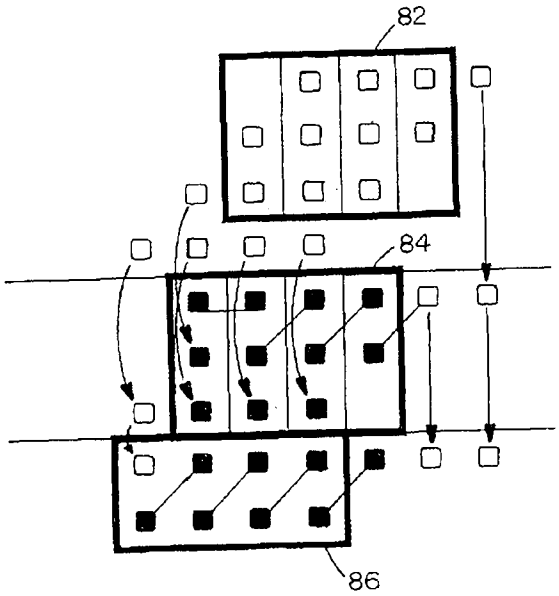


FIG. 9

【図１０】

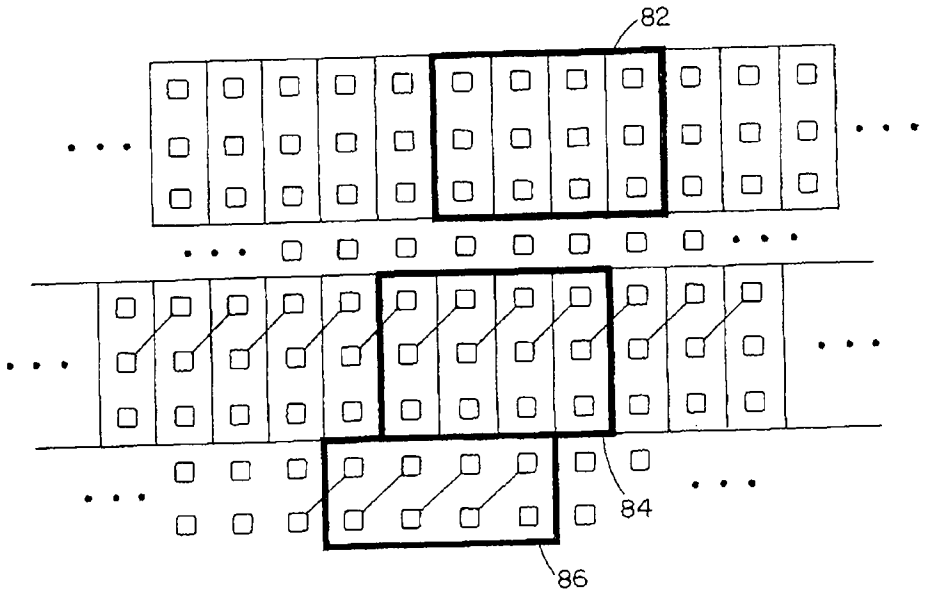
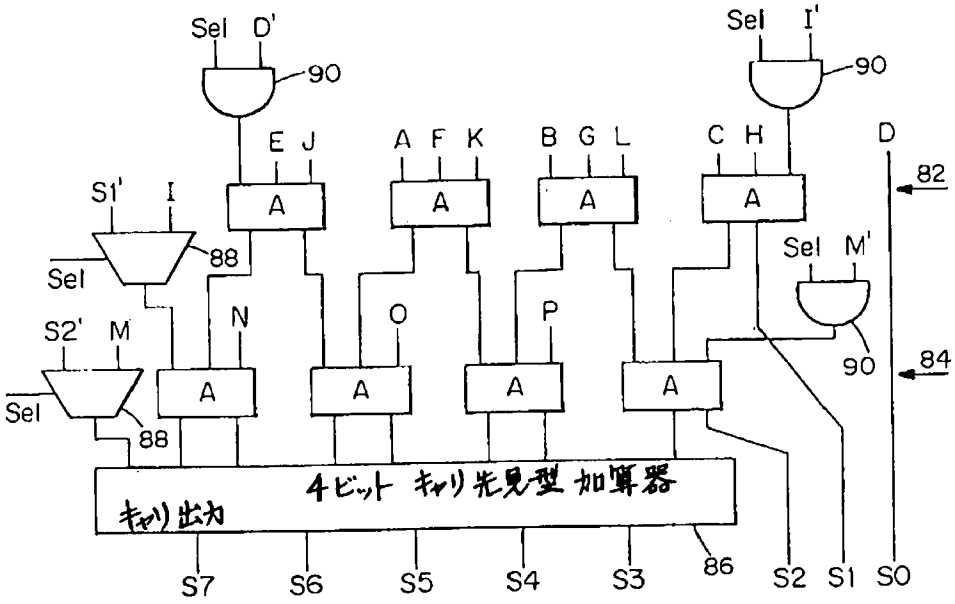


FIG. 10

【図１１】



【図１２】

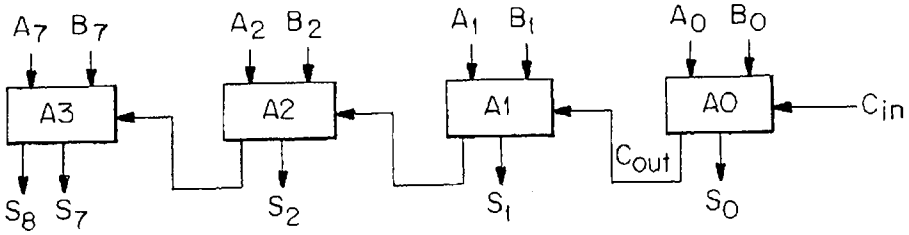


FIG. 12

【図１３】

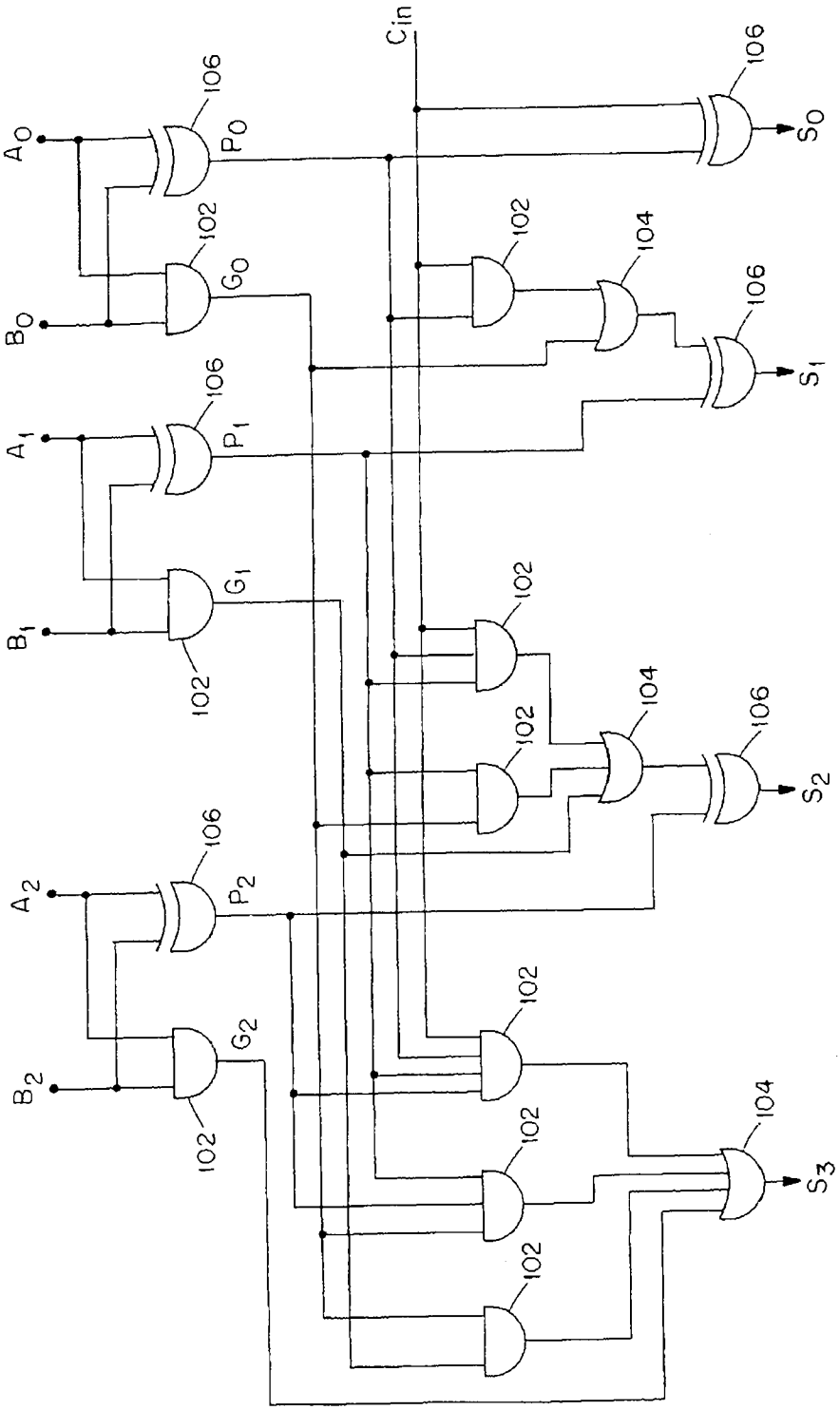
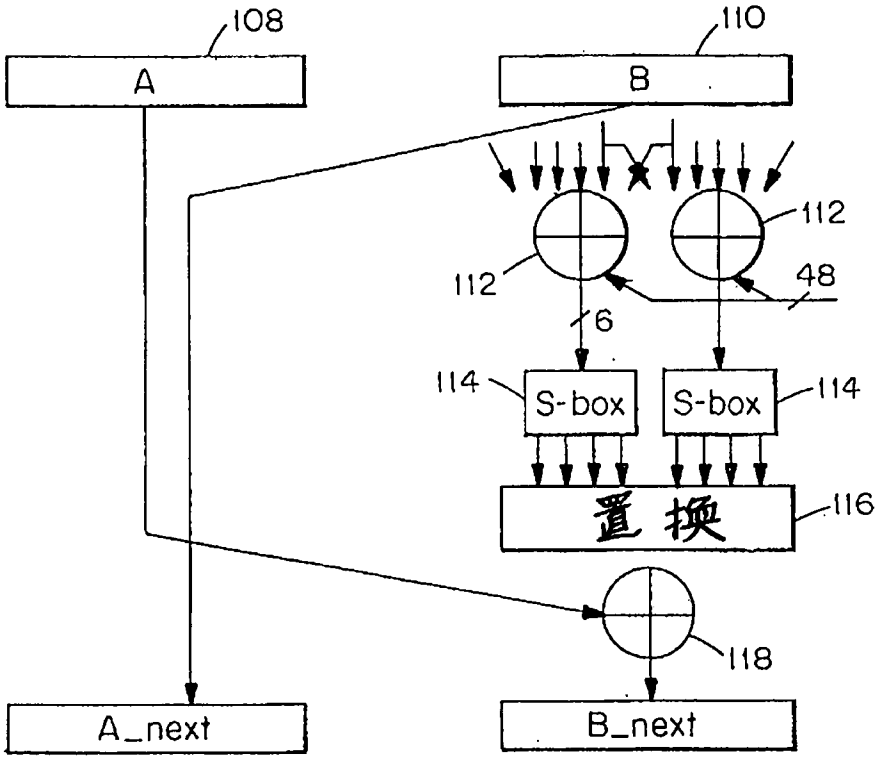
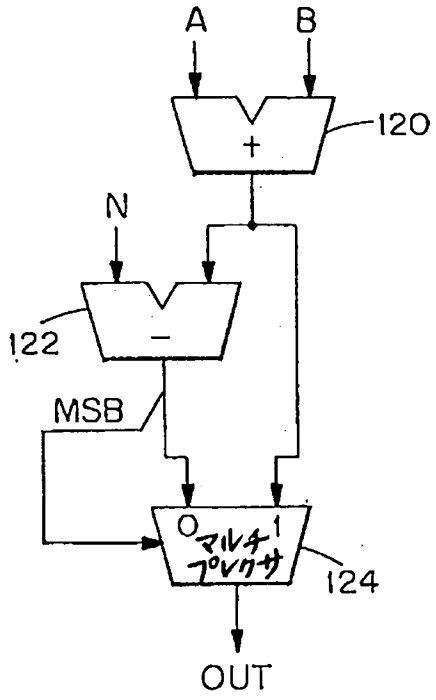


FIG. 13

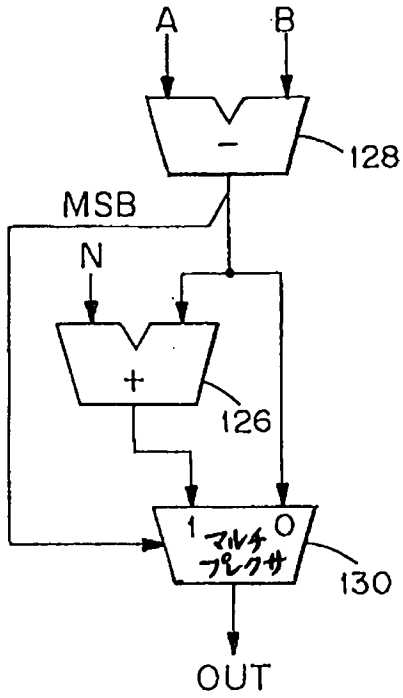
【図１４】


108
A
110
B
112
112
48
6
114
S-box
S-box
114
置換
116
118
A_next
B_next

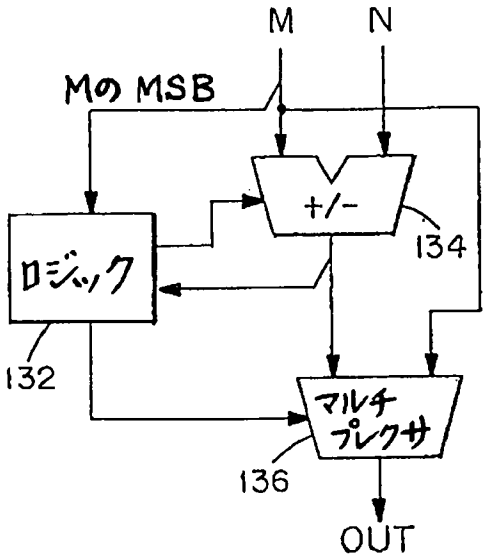
【図１５Ａ】


A
B
+
120
N
−
122
MSB
0
1
マルチプレクサ
124
OUT

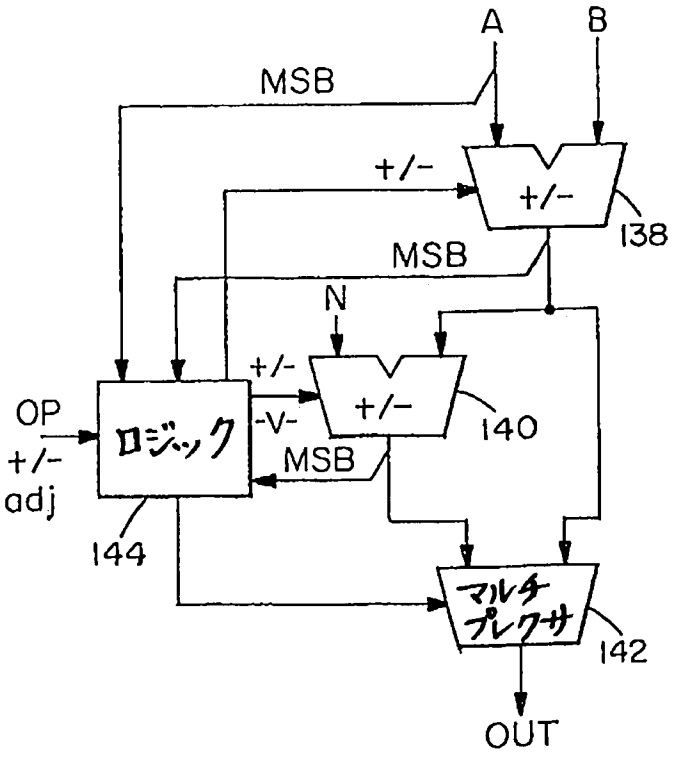
【図１５Ｂ】


A
B
−
128
MSB
N
+
126
1
マルチプレクサ
0
130
OUT

【図１５Ｃ】


M
N
Mの MSB
+/−
134
ロジック
132
マルチプレクサ
136
OUT

【図１５Ｄ】



【手続補正書】特許協力条約第３４条補正の翻訳文提出書
【提出日】平成１２年６月２７日（２０００．６．２７）
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】特許請求の範囲
【補正方法】変更
【補正内容】
【特許請求の範囲】
【請求項１】　単一のチップ上に演算処理装置（３７）のアレイを含む、電子暗号化デバイスであって、各演算処理装置は、
暗号化アルゴリズムのラウンドを記憶するための命令メモリ（６２）を含み、該ラウンドは命令のシーケンスを含み、各演算処理装置はさらに、
命令メモリからのラウンドを実現するためのプロセッサ（５６）と、
暗号化データオペランドおよび該ラウンドの実行によって得られた暗号化されたデータを記憶するためのデータ記憶装置（６６、６８）とを含み、
該アレイの演算処理装置は各々、ラウンドのうち１つを実現してその結果を連続する演算処理装置に転送し、それにより、該演算処理装置のアレイは演算処理装置パイプラインにおいて暗号化アルゴリズムの連続的なラウンドを実現する、電子暗号化デバイス。
【請求項２】　該データ記憶装置は、その１部分が、該線形アレイの隣接する演算処理装置間でデータを転送するために該線形アレイの隣接する演算処理装置間で共用される、請求項１に記載の電子暗号化デバイス。
【請求項３】　各演算処理装置は制御ユニット（６０）およびＡＬＵ（５６）を含み、該制御ユニット（６０）、命令メモリ（６２）およびデータ記憶装置（６６、６８）はローカル演算処理装置データバス（７２）に接続され、該ローカルデータバスはスイッチ（７６）によって２つの独立した区分に区分けされ、該制御ユニットおよび命令メモリは該区分のうちの一方に接続され、該ＡＬＵおよびデータ記憶装置は該区分のうち他方に接続される、請求項２に記載の電子暗号化デバイス。

【請求項４】　各演算処理装置は制御ユニット（６０）およびＡＬＵ（５６）を含み、該制御ユニット（６０）、命令メモリ（６２）、ローカルデータメモリ（６４）および共用データ記憶装置（６６、６８）はローカル演算処理装置バス（７２）に接続され、該ローカルバスはスイッチ（７６）によって、該命令メモリおよび該制御ユニットを接続するローカル命令バス区分と、該ＡＬＵ、ローカルデータメモリおよび共用データ記憶装置を接続するローカルデータバス区分とに区分けされ、該スイッチは、該２つのローカルバス区分上で独立した同時動作を可能にするか、または、該２つのバス区分間の通信を可能にする、請求項２に記載の電子暗号化デバイス。

【請求項５】　各演算処理装置は、該演算処理装置内で乗算演算を行なうための乗算器（７０）をさらに含む、請求項４に記載の電子暗号化デバイス。

【請求項６】　該暗号化アルゴリズムの実現中に、該パイプライン内の各演算処理装置は、結果として得られたデータを、後続の演算処理装置が直接アクセスすることができるように該後続の演算処理装置と共用されるデータ記憶装置（６８）内に書込む、請求項２に記載の電子暗号化デバイス。

【請求項７】　該演算処理装置の共用データ記憶装置（６６、６８）は、該線形アレイの隣接する演算処理装置間でデータを転送するために、該線形アレイの隣接する演算処理装置間で共用されるデュアルポートメモリで構成される、請求項２に記載の電子暗号化デバイス。

【請求項８】　各プロセッサは制御ユニット（６０）およびＡＬＵ（５６）を含み、該制御ユニット（６０）、ＡＬＵ、命令メモリ（６２）、ローカルデータメモリおよび共用データ記憶装置（６６、６８）はローカル演算処理装置データバス（７２）に接続され、該ローカルデータバスはスイッチ（７６）によって２つの独立した区分に区分けされ、該制御ユニットおよび命令メモリは該区分のうち一方に接続されかつ、該ＡＬＵ、ローカルデータメモリおよび共用データ記憶装置は該区分のうち他方に接続される、請求項７に記載の電子暗号化デバイス。

【請求項９】　各演算処理装置は、該演算処理装置内で乗算演算を行なうための乗算器（７０）をさらに含む、請求項１に記載の電子暗号化デバイス。

【請求項１０】　複数の演算処理装置の該乗算器は、より幅の広い乗算器の区分として連結されるように適合される、請求項９に記載の電子暗号化デバイス。

【請求項１１】　各乗算器は部分積加算器（Ａ）を含み、該加算器は、独立した乗算器として動作しているときには第１の入力の組を選択し、かつ、連結されているときには、隣接する演算処理装置からの入力を含む第２の入力の組を選択するための入力選択回路を有する、請求項１０に記載の電子暗号化デバイス。

【請求項１２】　各プロセッサは制御ユニット（６０）およびＡＬＵ（５６）を含み、該制御ユニット（６０）、ＡＬＵ、命令メモリ（６２）、ローカルデータメモリおよび共用データ記憶装置（６６、６８）はローカル演算処理装置データバス（７２）に接続され、該ローカルデータバスはスイッチ（７６）によって２つの独立した区分に区分けされ、該制御ユニットおよび命令メモリは該区分のうち一方に接続され、かつ該ＡＬＵ、ローカルデータメモリおよび共用データ記憶装置は該区分のうち他方に接続される、請求項１に記載の電子暗号化デバイス。

【請求項１３】　グローバルランダムアクセスメモリ（４４）およびグローバルバス（３８）をさらに含み、データは該グローバルランダムアクセスメモリと該演算処理装置データ記憶装置との間で該グローバルバスを通じて転送される、請求項１に記載の電子暗号化デバイス。

【請求項１４】　該グローバルバスに結合された、該演算処理装置によって処理されるデータ語よりも幅の広いデータ語を処理するための中央処理装置（４６）をさらに含む、請求項１３に記載の電子暗号化デバイス。

【請求項１５】　各演算処理装置は、該演算処理装置内で乗算演算を行なうための乗算器（７０）をさらに含む、請求項１４に記載の電子暗号化デバイス。

【請求項１６】　複数の演算処理装置の該乗算器は、より幅の広い乗算器の区分として連結されるよう適合される、請求項１５に記載の電子暗号化デバイス。

【請求項１７】　各乗算器は部分積加算器（Ａ）を含み、該加算器は、独立した乗算器として動作しているときには第１の入力の組を選択し、また、連結さ

れているときには隣接する演算処理装置からの入力を含む第２の入力の組を選択するための、入力選択回路を有する、請求項１６に記載の電子暗号化デバイス。

【請求項１８】　該中央処理装置は加算器を含み、該加算器は、

複数加算器区分（Ａ０～Ａ１５）を含み、該複数加算器区分の各々はキャリ出力および合計出力を有し、該複数加算器区分の各々は２つのオペランドのうち各オペランドの１区分を処理し、該加算器はさらに、

加算器サイクル内でキャリが得られる限り、連続的なクロックサイクル中、該キャリ出力を連続する加算器区分へのキャリ入力として選択するためのキャリ選択器（７９）と、

加算器サイクル内でキャリが得られる限り、連続的なクロックサイクル中、各合計出力を同じ加算器区分へのオペランド入力として選択するためのオペランド選択器（８０）とを含む、請求項１３に記載の電子暗号化デバイス。

【請求項１９】　各演算処理装置の各プロセッサは、ＭｍｏｄＮを計算するモジュロ調整演算を行なう、請求項１に記載の電子暗号化デバイス。

【請求項２０】　各演算処理装置の各プロセッサは、Ａ±ＢｍｏｄＮを計算するモジュロ加算または減算演算を行なう、請求項１に記載の電子暗号化デバイス。

【請求項２１】　各演算処理装置の各プロセッサは、Ａ×ＢｍｏｄＮを計算するモジュロ乗算演算を行なう、請求項１に記載の電子暗号化デバイス。

【請求項２２】　該暗号化デバイスは加算器をさらに含み、該加算器は、

複数加算器区分（Ａ０～Ａ１５）を含み、該複数加算器区分の各々は、キャリ出力および合計出力を含み、該複数加算器区分の各々は２つのオペランドのうち各オペランドの１区分を処理し、該加算器はさらに、

加算器サイクル内でキャリが得られる限り、連続的なクロックサイクル中、キャリ出力を連続する加算器区分へのキャリ入力として選択するキャリ選択器（７９）と、

加算器サイクル内でキャリが得られる限り、連続的なクロックサイクル中、各合計出力を同じ加算器区分へのオペランド入力として選択するオペランド選択器（８０）とを含む、請求項１に記載の電子暗号化デバイス。

【請求項２３】　単一チップ上に演算処理装置（３７）の線形アレイを含む、電子暗号化デバイスであって、各演算処理装置は、

暗号化アルゴリズムの少なくとも１つのラウンドを実現するのに必要とされるコードを記憶するための命令メモリ（６２）と、

該命令メモリからの該ラウンドを処理するためのプロセッサ（５６）と、

ローカルデータメモリ（６４）と、

２つの隣接する演算処理装置間の共用データ記憶装置（６６、６８）とを含み、

該線形アレイの演算処理装置は各々、該ラウンドのうち１つを実現しかつ、その結果を連続する演算処理装置に転送し、それにより、該演算処理装置の線形アレイは演算処理装置パイプラインにおいて該暗号化アルゴリズムの連続的なラウンドを処理する、電子暗号化デバイス。

【請求項２４】　該暗号化アルゴリズムの実現中、該パイプライン内の各演算処理装置は、結果として得られるデータを、後続の演算処理装置によって直接アクセスすることができるように該後続の演算処理装置と共用されるデータメモリ内に書込む、請求項２３に記載の電子暗号化デバイス。

【請求項２５】　演算処理装置（３７）の線形アレイを含む暗号化データ処理システムであって、各演算処理装置は、

命令メモリ（６２）と、

該命令メモリからの命令を処理するためのプロセッサ（５６）と、

データメモリ（６６、６８）とを含み、

該線形アレイの該演算処理装置のデータメモリは、該線形アレイの隣接する演算処理装置間でデータを転送するための、隣接する演算処理装置間で共用されるデュアルポートメモリを含む、暗号化データ処理システム。

【請求項２６】　各プロセッサは制御ユニット（６０）およびＡＬＵ（５６）を含み、該制御ユニット（６０）、ＡＬＵ、命令メモリ（６２）、および該演算処理装置のデータメモリは、ローカル演算処理装置データバス（７２）に接続され、該ローカルデータバスはスイッチ（７６）によって２つの独立した区分に区分けされ、該制御ユニットおよび命令メモリは該区分のうち一方に接続されか

つ、該ＡＬＵならびにローカルおよび共用データメモリは該区分のうち他方に接続される、請求項２５に記載の電子暗号化システム。

【請求項２７】　各演算処理装置は、該演算処理装置内で乗算演算を行なうための乗算器（７０）をさらに含む、請求項２５に記載の電子暗号化システム。

【請求項２８】　複数の演算処理装置の該乗算器は、幅のより広い乗算器の区分として連結されるように適合される、請求項２７に記載の電子暗号化システム。

【請求項２９】　各乗算器は部分積加算器（Ａ）を含み、該加算器は、独立した乗算器として動作しているときには第１の入力の組を選択し、また、連結されているときには隣接する演算処理装置からの入力を含む第２の入力の組を選択するための、入力選択回路を有する、請求項２８に記載の電子暗号化システム。

【請求項３０】　グローバルランダムアクセスメモリ（４４）およびグローバルバス（３８）をさらに含み、データは該グローバルランダムアクセスメモリと該演算処理装置データメモリとの間で該グローバルバスを通じて転送される、請求項２５に記載の電子暗号化システム。

【請求項３１】　該グローバルバスに結合されて、該演算処理装置によって処理されるデータ語よりも幅の広いデータ語を処理するための、中央処理装置（４６）をさらに含む、請求項３０に記載の電子暗号化システム。

【請求項３２】　該演算処理装置内で乗算演算を行なうための乗算器（７０）をさらに含む、請求項３１に記載の電子暗号化システム。

【請求項３３】　複数の演算処理装置の該乗算器は、より幅の広い乗算器の区分として連結されるように適合される、請求項３２に記載の電子暗号化システム。

【請求項３４】　各乗算器（７０）は部分積加算器（Ａ）を含み、該加算器は、独立した乗算器として動作しているときには第１の入力の組を、また、連結されているときには隣接した演算処理装置からの入力を含む第２の入力の組を選択するための、入力選択回路を有する、請求項３３に記載の電子暗号化システム。

【請求項３５】　該中央処理装置は加算器を含み、該加算器は、

複数加算器区分（Ａ０～Ａ１５）を含み、該複数加算器区分の各々はキャリ出力および合計出力を有し、該複数加算器区分の各々は２つのオペランドのうち各オペランドの１区分を処理し、該加算器はさらに、

加算器サイクル内でキャリが得られる限り、連続するクロックサイクル中、該キャリ出力を連続する加算器区分へのキャリ入力として選択するキャリ選択器（７９）と、

加算器サイクル内でキャリが得られる限り、連続するクロックサイクル中、各合計出力を同じ加算器区分へのオペランド入力として選択するオペランド選択器（８０）とを含む、請求項３１に記載の電子暗号化システム。

【請求項３６】　各演算処理装置の各プロセッサは、ＭｍｏｄＮを計算するモジュロ調整演算を行なう、請求項２５に記載の電子暗号化システム。

【請求項３７】　各演算処理装置の各プロセッサは、Ａ±ＢｍｏｄＮを計算するモジュロ加算または減算演算を行なう、請求項２５に記載の電子暗号化システム。

【請求項３８】　各演算処理装置の各プロセッサは、Ａ×ＢｍｏｄＮを計算するモジュロ乗算演算を行なう、請求項２５に記載の電子暗号化システム。

【請求項３９】　該暗号化デバイスは加算器をさらに含み、該加算器は、

複数の加算器区分（Ａ０～Ａ１５）を含み、該複数の加算器区分の各々はキャリ出力および合計出力を有し、該複数の加算器区分は２つのオペランドのうち各オペランドの１区分を処理し、該加算器はさらに、

加算器サイクル内でキャリが得られる限り、連続するクロックサイクル中、該キャリ出力を連続する加算器区分へのキャリ入力として選択する、キャリ選択器（７９）と、

加算器サイクル内でキャリが得られる限り、連続するクロックサイクル中、各合計出力を同じ加算器区分へのオペランド入力として選択する、オペランド選択器（８０）とを含む、請求項２５に記載の電子暗号化システム。

【請求項４０】　該暗号化アルゴリズムの実現中、該パイプライン内の各演算処理装置は、結果として得られたデータを、後続の演算処理装置が直接アクセスすることができるように該後続の演算処理装置と共用するデータメモリに書込

む、請求項２５に記載の電子暗号化デバイス。

【請求項４１】　乗算器回路であって、該回路は、

複数の乗算器区分を含み、その各々が第１の長さのオペランド語を受取り、さらに、

該乗算器区分が別個の乗算器として動作しているときには第１の入力の組を選択し、また、第２の語長のオペランドに対する演算を行なう幅のより広い乗算器として該乗算器区分を連結するためには第２の入力の組を選択する、入力選択器を含む、乗算器回路。

【請求項４２】　各乗算器区分は部分積加算器を含む、請求項４１に記載の乗算器。

【請求項４３】　電子暗号化デバイスであって、該デバイスは単一チップ上に、

演算処理装置（３７）の線形アレイを含み、その各々は、命令ストア（６２）と、データ記憶装置（６６、６８）と、該命令ストアからの命令のシーケンスを処理して第１の長さのデータ語に対する演算を行なうプロセッサ（５６）とを有し、該演算処理装置の該データ記憶装置は、該アレイの隣接する演算処理装置の間でデータを転送するために隣接する演算処理装置間で共用されるデュアルポートメモリを有し、該線形アレイの該演算処理装置は、自身の命令ストア内に、暗号化アルゴリズムのそれぞれのラウンドを記憶しかつ、該ラウンドの結果を連続する演算処理装置に転送し、よって、該演算処理装置の線形アレイは、演算処理装置パイプラインにおいて該暗号化アルゴリズムの連続するラウンドを処理し、さらに、

グローバルランダムアクセスメモリ（４４）と、

該グローバルランダムアクセスメモリと該演算処理装置データメモリとの間でそれを介してデータが転送される、グローバルバス（３８）と、

少なくとも該第１の長さよりも長い第２の長さのデータ語に対する演算を行なう、公開鍵暗号化プロセッサ（４６）とを含み、該公開鍵暗号化プロセッサは、該第２の長さの語長でグローバルランダムアクセスメモリにアクセスする、電子暗号化デバイス。

【請求項４４】　単一チップ上に演算処理装置（３７）のアレイを含む、電子暗号化デバイスであって、各演算処理装置は、

暗号化アルゴリズムのラウンドを記憶するための命令メモリ手段（６２）と、

該命令メモリからの該ラウンドを実現するためのプロセッサ手段（５６）と、

暗号化データオペランドおよび該ラウンドを実現することによって得られる暗号化されたデータを記憶するためのデータ記憶手段（６６、６８）とを含む、電子暗号化デバイス。

【請求項４５】　該データ記憶手段は、その一部が、該線形アレイの隣接する演算処理装置の間でデータを転送するために該線形アレイの隣接する演算処理装置の間で共用される、請求項４４に記載の電子暗号化デバイス。

【請求項４６】　グローバルランダムアクセス手段（４４）およびグローバルバス手段（３８）をさらに含み、該グローバルランダムアクセス手段と該演算処理装置データ記憶手段との間のデータの転送は該グローバルバス手段を介して行なわれる、請求項４５に記載の電子暗号化デバイス。

【請求項４７】　該グローバルバス手段に結合されて、該演算処理装置によって処理されるデータ語よりも幅の広いデータ語を処理するための、中央処理手段（４６）をさらに含む、請求項４６に記載の電子暗号化デバイス。

【請求項４８】　暗号化方法であって、該方法は、

単一チップ上の電子回路において、暗号化されるべきデータを受取るステップと、

該データを該チップ上のデータ演算処理装置のパイプラインに与えるステップとを含み、各演算処理装置は、暗号化のラウンドを処理し、その結果を連続する演算処理装置に転送し、それにより、該演算処理装置が演算処理装置のパイプラインにおいて該暗号化アルゴリズムの連続するラウンドを実現する、方法。

【請求項４９】　結果は共用メモリを介して連続する演算処理装置に転送される、請求項４８に記載の方法。

【請求項５０】　該チップ上の、グローバルバスを介して該演算処理装置に結合された中央処理装置で、暗号化アルゴリズムを処理するステップをさらに含み、該中央処理装置は、該演算処理装置によって処理されるデータ語よりも幅の

広いデータ語を処理する、請求項４９に記載の方法。

【請求項５１】　該チップ上の、グローバルバスを介して該演算処理装置に結合された中央処理装置において、暗号化アルゴリズムを処理するステップをさらに含み、該中央処理装置は、該演算処理装置によって処理されるデータ語よりも幅の広いデータ語を処理する、請求項４８に記載の方法。

【国際調査報告】

INTERNATIONAL SEARCH REPORT

Inte lonal Application No
PCT/CA 99/00176

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L9/06 H04L9/00 G06F15/80 G06F7/50 G06F7/52
G06F7/72

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	WO 89 07375 A (MOTOROLA INC) 10 August 1989 (1989-08-10) page 2, line 18 - line 23 page 4, line 1 - line 10 page 5, line 23 -page 6, line 9	1,23,45, 49 2,7,22, 24,25,50
Y A	US 4 922 418 A (DOLECEK QUENTIN E) 1 May 1990 (1990-05-01) column 3, line 41 - line 45 column 5, line 23 -column 6, line 3	2,7,24, 25,50 44

-/--

[X] Further documents are listed in the continuation of box C. [X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 November 1999	03. 12. 1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Verhoof, P

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte ional Application No
PCT/CA 99/00176

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WOLTER S ET AL: "ON THE VLSI IMPLEMENTATION OF THE INTERNATIONAL DATA ENCRYPTION ALGORITHM IDEA" 1995 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS), SEATTLE (US) APRIL 30 - MAY 3, 1995, vol. 1, 30 April 1995 (1995-04-30), pages 397-400, XP000583247 INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS page 397, right-hand column, line 18 - last line page 400, right-hand column, last paragraph	1,23,45, 49
A	WO 91 18460 A (TRAUTNER ROLF) 28 November 1991 (1991-11-28) page 13, last paragraph -page 15, line 6	1,23,25, 44,45,49
A	SAUERBREY J: "A MODULAR EXPONENTIATION UNIT BASED ON SYSTOLIC ARRAYS" ADVANCES IN CRYPTOLOGY - AUSCRPYT, GOLD COAST, QUEENSLAND, DEC. 13 - 16, 1992, no. CONF. 3, 13 December 1992 (1992-12-13), pages 505-516, XP000470468 SEBERRY J;YULIANG ZHENG the whole document	21,27
X	US 5 343 416 A (ROTSTAIN JEHOSHUA S ET AL) 30 August 1994 (1994-08-30)	41,42
A	the whole document	10,11, 16,17, 28,29, 33,34
X	EP 0 654 733 A (HEWLETT PACKARD CO) 24 May 1995 (1995-05-24)	41,42
A	figure 8	10,11, 16,17, 28,29, 33,34
X	US 3 098 153 A (H. HEIJN) 16 July 1963 (1963-07-16)	43
Y	column 6, line 50 - line 54; figures 3,6	22
A		18,35,39

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte...tional application No.
PCT/CA 99/00176

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☒ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☒ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International Application No. PCT/CA 99/00176

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-40, 44-52

Electronic encryption device comprising an array of processing elements, each processing element comprising an instruction memory and a processor for processing instructions from said memory as well as method for operating said encryption device.

2. Claims: 41, 42

Multiplier circuit comprising a plurality of multiplier segments and input selectors which select a first or a second set of inputs.

3. Claim : 43

Adder comprising plural adder segments, carry selectors and operand selectors.

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int[illegible]ional Application No
PCT/CA 99/00176

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 8907375	A	10-08-1989	US	4914697 A	03-04-1990
			AT	139392 T	15-06-1996
			CA	1336721 A	15-08-1995
			DE	68926670 D	18-07-1996
			DE	68926670 T	19-12-1996
			EP	0398931 A	28-11-1990
			HK	1004585 A	27-11-1998
			JP	3500117 T	10-01-1991
			KR	9614682 B	19-10-1996
US 4922418	A	01-05-1990	US	4720780 A	19-01-1988
			DE	3685107 A	04-06-1992
			EP	0237571 A	23-09-1987
			JP	63501530 T	09-06-1988
			KR	9701899 B	18-02-1997
			WO	8701841 A	26-03-1987
WO 9118460	A	28-11-1991	DE	4016203 A	21-11-1991
US 5343416	A	30-08-1994	NONE		
EP 0654733	A	24-05-1995	EP	0924601 A	23-06-1999
			JP	7200260 A	04-08-1995
			US	5636351 A	03-06-1997
US 3098153	A	16-07-1963	CH	363823 A	
			DE	1094020 B	
			FR	1192991 A	29-10-1959
			GB	876988 A	
			NL	98963 C	
			NL	213776 A	

フロントページの続き

(72)発明者 オコネル，コルマック・エム
カナダ、ケイ・２・ケイ １・ビィ・６
オンタリオ州、カナタ、ジャクソン・コート、27

Ｆターム(参考) 5B061 FF01 GG03 GG13 GG14 RR02
RR07
5J104 JA13 NA02

【要約の続き】